PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

TEMA:

MODELO PREDICTIVO FIDELIZACIÓN DE CLIENTES EN UNA EMPRESA DE TELECOMUNICACIONES

AUTOR:

MEJIA MEDINA BLANCA LUCIA

DIRECTOR:

ORTIZ NAVARRETE MIGUEL DIMITRI

TRABAJO DE TITULACIÓN PREVIA A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN SISTEMAS DE INFORMACIÓN CON MENCIÓN EN DATA SCIENCE

QUITO, 2024

DEDICATORIA

A Dios por darme las fuerzas necesarias para cumplir mis metas y levantarme en los momentos difíciles.

Con mucho amor y cariño dedico este proyecto mi familia que son el pilar fundamental y mi motor para cumplir con todas las metas que me propongo, gracias a las fuerzas y apoyo que me brindan en todas las etapas de mi vida.

A mis amados Padres, por ser mi constante inspiración, por su infinito amor y confianza. Gracias por su ejemplo y enseñarme que con esfuerzo, trabajo y dedicación todo es posible.

A mis hermanos y a mis sobrinos por su amor y ocurrencias que me alegran la vida.

AGRADECIMIENTO

A Dios, por su protección durante todos los días de mi vida.

Al Máster Miguel Ortiz por su apoyo, experiencia y por compartir sus conocimientos durante la elaboración del presente proyecto.

A todas las personas que me han brindado su apoyo y palabras oportunas para cumplir con esta meta personal y profesional.

RESUMEN

El objetivo principal del proyecto fue aplicar técnicas de clasificación, desarrollar modelos de aprendizaje automático para ayudar a entender y predecir el comportamiento de un cliente cuando pretende cancelar el servicio en una empresa de Telecomunicaciones.

Los modelos predictivos Regresión Logística, Árbol de decisión, Redes Neuronales, *Random Forest Classifier* y *Lazy Classifier* fueron desarrollados en Python, aplicando técnicas de clasificación de aprendizaje supervisado, empleando técnicas de *Data Mining* y todas las fases que compone el modelo CRISP-DM, en cada una de las etapas se desarrollaron acciones con el conjunto de datos usado como base para el proyecto, en cada una cada una se detallaron los resultados y hallazgos encontrados.

Al finalizar el desarrollo se han comparado los resultados de los diferentes modelos y la precisión de cada uno de ellos, arrojando que el modelo óptimo para este proyecto fue el de Regresión Logística. Al analizar todos los resultados, evaluar el modelo con mayor precisión se sugerirán a las jefaturas, marketing y todas las áreas se considere ingresar datos reales en los modelos predictivos y en base a los resultados se puedan personalizar campañas de promociones, optimizar los planes actualmente contratados y aplicar técnicas para garantizar la fidelizad de los clientes con la empresa.

ABSTRACT

The objective of the project was to apply classification techniques, develop machine learning models to help understand and predict the behavior of a customer when they intend to cancel the service in a Telecommunications company.

The predictive models Logistic Regression, Decision Tree, Neural Networks, Random Forest Classifier and Lazy Classifier were developed in Python, applying supervised learning classification techniques, using Data Mining techniques and all the phases that make up the CRISP-DM model, in In each of the stages, actions were developed with the set of data used as a basis for the project, in each one the results and findings found were detailed.

At the end of the development, the results of the different models and the precision of each of them were compared, showing that the optimal model for this project was Logistic Regression.

After analyzing all the results, evaluating the model with greater precision, it will be suggested to management, marketing and all areas, it will be considered to enter real data into the predictive models and based on the results, promotional campaigns can be personalized, current plans optimized and apply techniques to guarantee customer loyalty with the company.

INDICE

INDICE DE FIGURAS

INDICE DE TABLAS

1. Introducción

1.1. Generalidades

Los modelos predictivos permiten a las empresas medir los niveles de satisfacción con el servicio y la atención al cliente, también predecir cuando un cliente muestre señales o indicios de inconformidad con el servicio y una posible cancelación. Las empresas para retener a los clientes y mantener el contrato activo pueden ofrecer mejoras, realizar lanzamientos de nuevas promociones, bonificaciones personalizadas agradables y atractivas para el cliente.

Como herramienta de ayuda es importante disponer de modelos predictivos que anticipen el accionar y la decisión de los clientes cuando piensan cancelar el servicio. Otra importante utilidad de los modelos predictivos es la reducción de riesgos y fraudes, ya que se puede analizar posibles clientes sospechosos.

Debido al alto índice de clientes que cancelan el servicio y las múltiples quejas que se reciben a diario en una empresa de Telecomunicaciones, se plantea elaborar un modelo predictivo para conocer y predecir la fidelidad de los clientes en la empresa, evitar las posibles cancelaciones que puedan existir en base a las tendencias y el comportamiento que presentan previamente los clientes.

Los modelos predictivos son muy esenciales en las empresas y corporaciones, gracias a ellos, en base a los indicadores, resultados y estadísticas que se obtengan, los directivos y jefaturas pueden tomar decisiones de mejora, optimización, prescindir de ciertos procesos o procedimientos que sean antiguos y que ya no estén alineados con el giro de negocio. Los modelos predictivos ayudan a las áreas de marketing y publicidad a mantenerse actualizadas con las últimas tendencias e influir en las decisiones de los clientes, proporcionando promociones, beneficios personalizados de retención, entre otras ventajas.

1.2. Planteamiento del problema

En una empresa de Telecomunicaciones se ha observado que los clientes generan constantemente datos de entrada en base a los requerimientos, reclamos, asesorías, consultas, entre otras formas de comunicación con la compañía. Se ha evidenciado que la entrada de datos es gigante y darles uso es fundamental en la empresa, además de tener almacenados grandes cantidades de datos importantes, los mismos se pueden analizar para la toma de mejores decisiones. Se aspira realizar un modelo predictivo para determinar el comportamiento del cliente que pretende cancelar los servicios de

Internet, Tv, Telefonía y aplicaciones de *streaming* en una empresa de Telecomunicaciones.

1.3. Objetivos

1.3.1. Objetivo General

Aplicar modelos de predicción para identificar el comportamiento de los clientes antes que soliciten la cancelación del servicio usando un modelo de aprendizaje automático para reducir la tasa de cancelación de servicios en una empresa de Telecomunicaciones.

1.3.2. Objetivos Específicos

- Explicar el funcionamiento de los modelos predictivos a través de aprendizaje automático mediante de una sólida fundamentación de las leyes y teorías que sustentan la misma.

- Aplicar la metodología de CRISP-DM para predecir la cantidad de clientes que cancelan el servicio.

- Diseñar un modelo predictivo partir de los datos particulares de quejas del usuario que prediga los clientes que van a cancelar el servicio contratado.

- Describir y determinar el comportamiento y las tendencias que presentan los clientes antes de realizar la cancelación del servicio.

1.4. Justificación

La empresa de Telecomunicaciones a la que está enfocada el desarrollo de este Proyecto de Titulación se dedica a brindar servicios de internet, telefonía, Tv y aplicaciones de *streaming* a clientes nivel nacional.

Es importante investigar sobre este tema ya que en las empresas de telecomunicaciones las ganancias y la posición en el mercado dependen de la acogida y el nivel de satisfacción que tienen los clientes. Los clientes a través de sus recomendaciones pueden atraer más clientes para que contraten los servicios que la empresa proporciona.

Otra razón, es porque el cliente está adquiriendo y pagando por un servicio y espera que sea óptimo y de alta calidad, pues a un cliente, le gusta un servicio de alta velocidad, que sea estable, que no presente fallas e interferencias.

1.5. Alcance

Se pretende realizar los modelos predictivos Regresión Logística, Árbol de decisión, Redes Neuronales, *Random Forest Classifier* y *Lazy Classifier* para poder determinar un comportamiento del cliente que pretende cancelar los servicios en una empresa de Telecomunicaciones. Con los datos que hayan recolectado en el punto anterior, se procederá a desarrollar los modelos predictivos.

Se ha seleccionado este tema ya que, al trabajar en el área de Servicios de TI de una empresa de Telecomunicaciones, a diario se reciben quejas e inconformidades por parte de los clientes con respecto a temas de velocidad, cobertura, entre otras. Se elaborarán los modelos predictivos para fidelización de clientes con un dataset gratuito, posteriormente y ya con el conocimiento adquirido en este proyecto de titulación realizar un modelo predictivo real en la empresa.

El motivo para realizar este modelo predictivo es precisamente ayudar en el área de la empresa donde laboro a proporcionar técnicas y métodos para poder mitigar y brindar mejoras de servicio, brindar promociones, apuntar con nuevas alternativas y estrategias de mejora continua, para lograr retener a los clientes y que, al contrario, en vez de que cancelen el servicio, ellos recomienden y traigan a más clientes.

2. Revisión Literaria

En este capítulo se abordarán los principales conceptos, teorías más relevantes e importantes de todo lo relacionado con *Machine Learning* o aprendizaje automático, Inteligencia Artificial, Big Data y tipos de clasificaciones de los distintos tipos de aprendizaje y metodologías aplicadas a patrones que permiten realizar minería de datos.

La data tomada como base para realizar el presente proyecto es gratuita, está basada en las características similares de los clientes de la empresa, poseen valores similares en cuanto a código único, género, si el cliente tiene dependientes, si el cliente tiene pareja, si pertenece a la tercera edad, los años de antigüedad en la empresa, los servicios que posee, la forma de pago, entre otras características relevantes.

2.1. Machine Learning

Machine Learning o Aprendizaje Automático en español, es la ciencia del subcampo de la Inteligencia Artificial del mundo de la computación que ha ido evolucionando y dando un giro de negocio en las empresas e instituciones en los últimos años, ha

transformado la forma de trabajar y la lógica de negocio gracias al uso de la tecnología.

Machine Learning se puede definir como la implementación de algoritmos para optimizar y automatizar modelos informáticos tradicionales, lo que permite entrenar a un modelo informático para que realice las acciones parecidas a las del ser humano, las instrucciones y funcionamiento dependen totalmente de las reglas, parámetros e hiperparametros.

Machine Learning también se complementa con el mundo del Big Data, ya que permite trabajar con grandes volúmenes de datos a través de su extracción de los principales servidores y fuentes de información de las empresas. Para realizar modelos de aprendizaje con una alta precisión es importante tener gran número de datos y muestras, el éxito de los modelos dependerá al 100% del correcto procesamiento de la información y el óptimo tratamiento de los outliers o valores atípicos que se pueden presentar en el conjunto de datos.

El Aprendizaje Automatizado (Machine Learning; ML) es una rama de la inteligencia artificial, en gran parte inspirada en el razonamiento humano, que comprende el aprendizaje a partir de experiencia. (Sammut y Webb 2011).

El aprendizaje automático aborda, a su vez, una serie de problemáticas que tributan a problemas específicos, entre ellos: los problemas de clasificación, asociación, agrupamiento, y selección de rasgos. (Sammut y Webb 2011).

Para obtener el conocimiento y la experiencia necesarios para una acertada toma de decisiones en la empresa, en el siguiente proyecto se desarrollarán varios tipos de modelos predictivos y la aplicación de algoritmos de clasificación, como Regresión Logística, Árbol de decisión, Redes Neuronales, Random Forest Classifier y Lazy Classifier. Con el amplio conocimiento del funcionamiento de los modelos y con los resultados del modelo más relevante se podrán tomar decisiones respecto cambios de estrategias, nuevas campañas de marketing, promociones personalizadas y otro tipo de retenciones para los clientes.

2.1.1. Principales campos de acción en donde se usa Machine Learning e Inteligencia Artificial

- Finanzas: Se usa para detectar fraudes, clientes elegibles para otorgarles prestamos, ofrecer productos y servicios y de igual manera para conocer la fiabilidad que tiene un cliente para pagar sus créditos.

- Área de Marketing: Se puede decir que esta área es una de las que más consumen y recurren al uso de modelos predictivos y aprendizaje automático, se basa en la selección de clientes gracias a la técnica conocida como *clustering* o agrupamiento por sectores para ofrecer productos o servicios dependiendo el lugar de residencia, edades, temporadas tales como regreso a clases, temporadas de campeonatos deportivos, temporadas navideñas, fin de año, etc.

- Empresas de Telecomunicaciones: En estas empresas también es muy popular, gracias a los modelos de predicción automática se puede detectar el comportamiento de los clientes y en base a los resultados se pueden ofrecer beneficios, tales como, nuevas promociones para retener a los clientes, mejoras de planes continuos en base a las nuevas tecnologías de internet que van apareciendo en el mercado, promociones por sectores y precios especiales, promociones por temporadas de inicio escolar, promociones y descuentos a clientes que poseen carnet de Conadis y Tercera edad, beneficios por fechas especiales, etc.

- Medicina: En los programas que usan los médicos es muy conveniente para mantener el historial y los cuadros de salud ha presentado un paciente y conocer todas sus enfermedades, tratamientos realizados, patologías con el objetivo de suministrar el tratamiento adecuado y no involucrar ni comprometer la salud y hasta la vida de un paciente.

- Educación: En los establecimientos es muy importante para conocer el porcentaje de estudiantes que desertan las carreras, los estudiantes que se retiran de las instituciones, los que tienen probabilidades de aprobar sin inconvenientes, análisis económicos para decidir proporcionarle una beca de estudios o de deportes, descuentos por pronto pago, precios al pagar en efectivo o con tarjeta de crédito la matricula y respectivos créditos, entre otros.

- Clima: Se usa mucho para poder predecir el clima y la temperatura que existirá en la ciudad, en base al cálculo y a las condiciones atmosféricas se puede estimar el comportamiento que tendrá el clima en un determinado día.

- Reconocimiento facial: En la actualidad es muy común encontrar puertas y sistemas biométricos que funcionan con un sistema de reconocimiento facial en donde el empleado tendrá que pararse frente a una cámara y se activará el ingreso en base a los patrones y los rasgos faciales de un empleado.

- Lo mismo ocurre en las aplicaciones bancarias, en donde la gran mayoría ya han

implementado el uso de reconocimiento facial, la fotografía e información ingresada por el cliente se conectará con la base de datos que se encuentra almacenada en el registro civil y se podrá determinar si es la persona en cuestión u otra persona, de esta manera se pueden evitar fraudes y posibles robos y actos ilícitos.

- Fidelización de clientes: El desarrollo de este proyecto ayudará a identificar los clientes que posiblemente cancelen el servicio, en base a los avisos previos que indique el modelo predictivo, se pueden identificar de manera oportuna los clientes que piensan cancelar el servicio y a partir de los resultados se puede ofertar planes, servicios personalizados, nuevas promociones y planes de retención a los clientes.

2.2. Aprendizaje Supervisado

En el aprendizaje supervisado, el agente observa pares de datos de entrada y salida a modo de ejemplo para aprender una función que modele la salida según la entrada (Russell & Norvig, 2010).

En este tipo de aprendizaje se usan conjuntos de datos etiquetados (labels – etiquetas) para separar y clasificar los datos y predecir los modelos con precisión. Los modelos se entrenan con los datos etiquetados y al ingresar nuevos datos de un tipo similar permitirá garantizar la exactitud de las predicciones del modelo y conocer el comportamiento esperado.

Dependiendo los datos de entrenamiento previos se podrá inferir en un nuevo resultado, por ejemplo, en base al mayor número de etiquetas se podrá calcular y determinar el nuevo valor, este modelo se usa mucho para calcular precios de casas, diagnóstico de una enfermedad en base al número y tipo de síntomas, en el precio de un celular basado en los precios de los meses similares a los del año pasados y otras funciones.

2.3. Big Data

La mayoría de las definiciones de big data están asociadas al volumen de los datos en términos de almacenamiento. Sin embargo, big data no es solo volumen, características tales como variedad y velocidad son igualmente importantes. Russom Philip (2011).

Big Data surge como una nueva era en la exploración y utilización de datos. Desde la perspectiva empresarial Big Data no representa solo grandes volúmenes de datos, se deben considerar los patrones extraídos a partir de los datos y que pueden generar

procesos de innovación. P. Zikopoulos and C. Eaton (2011).

2.4. Minería de Datos (Data Mining)

Actualmente las empresas manejan grandes cantidades de datos, por lo cual deben ser adaptativas a los cambios, aprender como generar nuevas estrategias de mercado, nuevas ventas y para ello es muy importante que existan profesionales capaces de identificar qué tipo de conocimiento e información están proporcionando sus datos, en base al análisis obtenido se pueden establecer nuevos estándares y modelos de negocio con los datos ya convertidos en información y con resultados reales.

Uno de los beneficios es que ayuda a predecir comportamientos, reducir costos, recursos, riesgos de perdidas, nuevos posicionamientos en el mercado, predecir comportamientos, generar nuevas estrategias de marketing, lograr la difusión a través de las redes sociales y del comercio electrónico.

Las organizaciones, en la búsqueda por la obtención de los mejores resultados de su gestión organizacional, adoptan la flexibilización como estrategia, con el objetivo de adecuarse a un mercado globalizado, dando origen a un proceso que incide en su sistema estructural. Así pues, una empresa flexible es la que se orienta hacia los clientes, posee tecnología nueva y presenta acuerdos laterales de organización e innovación. (Hansen y Mouritsen, 1999).

En el desarrollo del proyecto se utilizarán herramientas de minería de datos para el desarrollo, evaluar la eficacia y precisión del modelo, de esta manera se identificarán patrones de comportamiento de clientes que pretenden cancelar los servicios y para identificar de manera más clara se desarrollarán modelos predictivos para anticipar el comportamiento de un cliente antes de la cancelación.

2.4.1. Técnicas de Minería de Datos

Las técnicas de minería de datos dependen de las necesidades, objetivos y metas que tengan las empresas, dependiendo aquello se decidirá cuál es la mejor técnica para aplicarse para un determinado proceso.

Técnicas de Clasificación: Clasifica los productos en clases predeterminadas en función de parámetros y categorías importantes para la empresa, en base a ellos se segmentan por categorías de clientes, el objetivo es ofertar servicios personalizados para garantizar la fidelización del cliente y la satisfacción con el servicio.

La clasificación es el proceso de encontrar un modelo (o función) que describa y distinga clases de datos o conceptos. El modelo se deriva del análisis de un conjunto de datos de entrenamiento (es decir, objetos de datos para los cuales se conocen las etiquetas de clase). El modelo se utiliza para predecir la etiqueta de clase de objetos cuya etiqueta de clase se desconoce. (Jiawei Han,2006).

2.4.2. Modelo de Minería de Datos

Debido a todos los usos que se puede realizar con la minería de datos existen distintos modelos y metodologías para aplicarlas de acuerdo con los criterios y necesidades de las empresas y en base al conjunto de datos existente.

2.4.3. CRISP-DM

CRISP-DM establece un conjunto de tareas definidas en cuatros niveles de abstracción (fases, tareas generales, tareas específicas e instancias del proceso), que están estructuradas de forma jerárquica, iniciando desde el nivel general hasta el nivel específico. Según (Chapman et al., 2000).

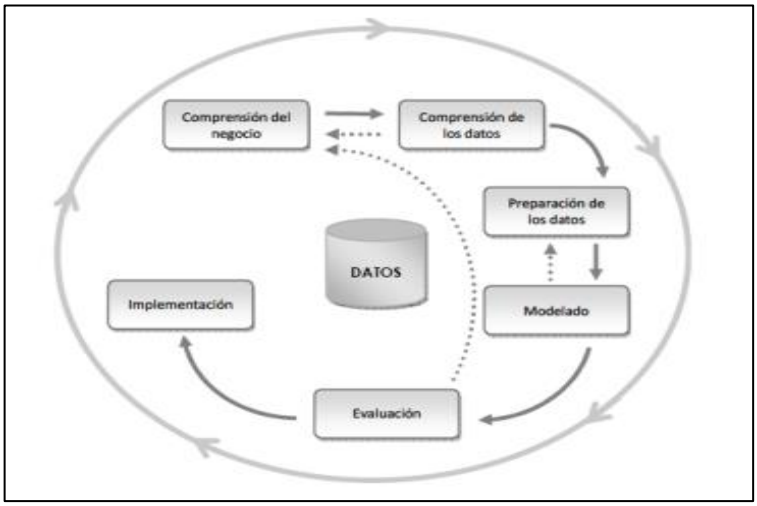


Figura 1: Fases CRISP-DM. Autor: Chaptman, 2000.

En la **Figura 1** se ha decidido aplicar la metodología CRISP-DM porque es iterativa y se puede regresar a cualquier fase en caso de necesitar ajustes en el conjunto de datos, realizar normalización y tratamiento adecuado de los datos. Permite trabajar con grandes volúmenes de datos, tener una estructura clara, está orientada al negocio, sus fases son claras y muy detalladas desde el inicio que empieza por la comprensión del negocio hasta el final que es la implementación.

Se puede tener grupos de varias áreas de interés y cada integrante del equipo aportará con sus conocimientos, convirtiendo en un trabajo colaborativo y homogéneo. Finalmente permite al negocio comprender sus metas e intereses desde el inicio proporcionando herramientas para la gestión, planificación y seguimiento del proyecto en todas sus fases. La metodología CRISP-DM está comprendida en seis fases que se detallan a continuación:

- Entendimiento del negocio
- Entendimiento de los datos
- Preparación de datos
- Modelado
- Evaluación
- Implementación

3. Resultados

La empresa de Telecomunicaciones por su línea de negocio dispone de una gran cantidad de datos de clientes, darles utilidad es fundamental. Los datos convertidos en información se vuelven oro puro, a través de los modelos predictivos se puede evidenciar el alcance, número de ventas, nuevos clientes y lo que analizaré en este proyecto es específicamente es los clientes que cancelan el contrato y desertan de los servicios de la empresa.

Por las características que presenta el modelo predictivo, por su gran cantidad de datos y por las etapas que atravesará se usará la metodología CRISP-DM, una ventaja es que el CRISP-DM permite regresar a la etapa anterior las veces que sean necesarias hasta encontrar el mejor modelo y algoritmo. La metodología CRISP-DM está comprendida en seis etapas o fases que se detallan a continuación:

3.1. Entendimiento del negocio

La fase inicial pone enfoque en los objetivos del proyecto y los requerimientos desde la perspectiva del negocio, convierte este conocimiento en la definición de un problema de minería de datos y un plan preliminar para lograr objetivos.

En las empresas de Telecomunicaciones es fundamental ofrecer servicios de calidad y cumplir con las expectativas del cliente con respecto a temas de velocidad, precio, accesibilidad y cobertura de servicios de internet, telefonía, televisión y aplicaciones.

La fidelización de clientes es un proceso que tiene por objetivo desarrollar una sólida relación entre las clientes y la empresa, las funciones que ayudan a la empresa a crecer son: la contratación de nuevos servicios, recomendación del servicio, crear alianzas duraderas entre el cliente y la empresa, mismas que significan ganancias y aumento de la cartera de la empresa.

Otra forma en la que las empresas buscan alternativas de mantener lazos con los clientes es a través de retenciones, se ha demostrado que ofrecer promociones personalizadas, aumento de velocidad, reducción en sus facturas con mejores beneficios asegura que el cliente mantenga los servicios y el contrato con la empresa. La empresa de Telecomunicaciones a menudo se encuentra diseñando, elaborando campañas de Marketing y promociones para cumplir con las expectativas del cliente, tener ganancias, ganar estabilidad y reconocimiento en el mercado.

Misión: "Brindar servicios de telecomunicaciones innovadores y confiables que conecten a las personas, empresas y comunidades, mejorando la calidad de vida a través de la tecnología y el compromiso con la excelencia en el servicio al cliente."

Visión: "Ser líderes en el sector de telecomunicaciones, reconocidos por nuestra capacidad para anticipar las necesidades de nuestros clientes y ofrecer soluciones tecnológicas vanguardistas. Aspiramos a ser la elección preferida de los clientes, destacando por nuestra innovación, confiabilidad y compromiso con la comunidad".

3.1.1. Organización del Negocio

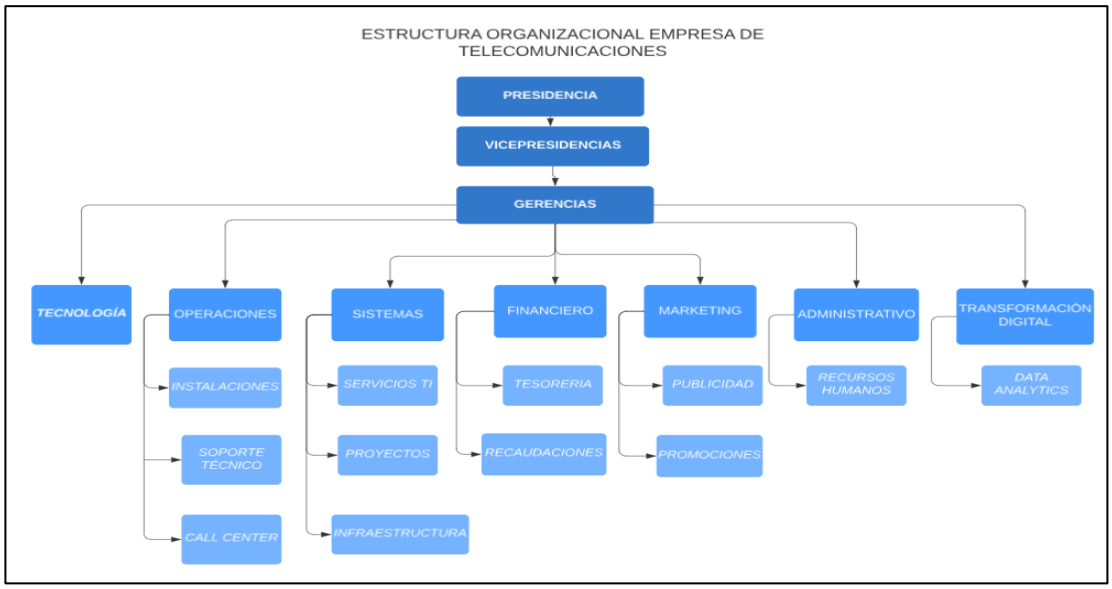


Figura 2: Organigrama Empresa de Telecomunicaciones. Autor: Blanca Mejía, 2024.

En la **Figura 2** se describe la estructura organizativa de la empresa, la jerarquía y las relaciones entre los diferentes departamentos y cargos. Es la representación visual y conceptual de cómo se organiza las autoridades y las relaciones de subordinación en la empresa.

3.1.2. Problemática por resolver

Se ha descrito la importancia de mantener satisfecho al cliente con el servicio que ofrece la empresa de Telecomunicaciones, sin embargo, no en todos los casos todos esos puntos pueden cumplirse, en la empresa en la cual laboro y de la que he hecho el análisis y comprensión del negocio, existen varios motivos por los cuales el cliente no se encuentra conforme con el servicio y prefiere cancelarlo.

La retención de clientes es radical para el crecimiento de la empresa y su posicionamiento en el mercado. Al comprender y abordar las razones detrás de las cancelaciones, se espera mejorar la satisfacción del cliente, reducir la pérdida y fuga de clientes, generar planes de retención y posible recomendación del servicio.

Los motivos más recurrentes y determinantes para que un cliente cancele el servicio son, la cobertura, velocidad, precios, el tipo de atención y tiempo de resolución de problemas. Pensando en todas las necesidades del cliente, se ha propuesto implementar un modelo de predictivo de clasificación y con la metodología CRIS-DM para conocer los escenarios que muestra el cliente antes de cancelar los servicios, este modelo de fidelización de clientes permitirá anticiparse a las cancelaciones, establecer estrategias de fidelización y retención.

3.1.3. Objetivos del negocio

Buscar satisfacer las necesidades de los clientes a través de servicios de calidad basados en tecnología de punta y un talento humano altamente capacitado y con vocación de servicio. En este proyecto se plantea reducir la tasa de cancelación de servicios en una empresa de Telecomunicaciones.

3.1.4. Criterio de éxito

El modelo debe anticipar la cancelación de servicios de un cliente, ya sea con los avisos y llamadas que un cliente realiza a la empresa, comprendiendo de mejor manera los puntos débiles, los aspectos en donde se deben mejorar, optimizar los procesos y tiempos de atención, para ello el modelo predictivo ayudará a comprender estas señales.

Aumento de la satisfacción del cliente: El modelo debe ser fácil de usar, entender las necesidades que tiene el cliente por parte de las áreas encargadas de realizar medición e indicadores. Estas áreas pueden ser Publicidad, Marketing, Sistemas, Jefaturas, entre otras. Entendiendo estos parámetros se podrá realizar mejoras, evitar la pérdida y deserción de clientes.

Aumento de la precisión de la medición de la deserción de los clientes: El modelo debe aumentar la precisión de deserción de clientes, lo que permitirá a la empresa tomar mejores decisiones con respecto a promociones, ofertas, servicios, nuevos planes, entre otros.

Mejora de la comprensión de las necesidades del cliente: El modelo debe ayudar a la empresa a comprender mejor el comportamiento de los clientes, lo que les permitirá desarrollar programas, políticas más eficaces para reducir la deserción de clientes y afianzar los convenios con el mismo. Estos criterios están alineados con los objetivos del negocio del proyecto, que son desarrollar un modelo predictivo que sea preciso, eficaz, entendible y fácil de interpretar por las personas responsables de analizar el comportamiento de los clientes.

3.1.5. Evaluación de la situación actual

Inventario de recursos

a. **Recursos humanos**

Equipo de proyecto: El equipo de proyecto debe estar formado por personas con experiencia en los siguientes campos:

- Minería de datos.
- Estadística.
- Marketing.
- Expertos en la materia: El equipo de proyecto debe trabajar en estrecha colaboración con expertos en la materia para comprender los desafíos del análisis de datos, comprensión del negocio, nuevas tendencias, tecnologías y otros campos correspondientes.

b. Recursos técnicos

- Base de Datos: La base de datos de los clientes es una fuente importante de datos para el proyecto.

- Herramientas de minería de datos: El equipo que se encuentre realizando el proyecto utilizará herramientas de minería de datos para desarrollar y evaluar el modelo predictivo.

- Herramientas de visualización de datos: El equipo que desarrolla el proyecto utilizará herramientas de visualización de datos para ayudar a los *stakeholders* (grupos de interés) a comprender los resultados del modelo.

c. Recursos financieros

Financiamiento: La elaboración del presente proyecto no es financiado por la empresa, por lo que el *dataset* es tomado de un sitio gratuito y los recursos son propios.

d. Recursos temporales

Duración del proyecto: El proyecto debe tener una duración definida para garantizar que se cumplan los objetivos del negocio. La duración de este proyecto será de tres meses, en el mismo se realizarán todas las fases de la Metodología CRISP-DM, detallando una a una con sus respectivos pasos y sub-fases.

e. Recursos físicos

Espacio de trabajo: El equipo de proyecto necesitará espacio de trabajo para desarrollar y evaluar el modelo.

3.1.6. Requisitos, supuestos y restricciones

a. Requisitos

El modelo debe ser preciso: El modelo debe ser capaz de clasificar correctamente a los clientes que piensan cancelar el servicio con la empresa de Telecomunicaciones.

El modelo debe ser eficaz: El modelo debe proporcionar información útil para tomar medidas para implementar puntos de mejora y políticas de retención al cliente.

El modelo debe ser utilizable: El modelo debe ser fácil de usar y entender por parte de los *stakeholders*. La información debe estar actualizada y disponible en todo momento.

b. Supuestos

La base de datos "Data Telecomunicaciones" es fiable: La base de datos debe proporcionar datos precisos y completos.

Las técnicas de minería de datos utilizadas son eficaces: Las técnicas de minería de datos utilizadas deben ser capaces de identificar patrones en los datos que pueden ayudar a predecir la deserción de un cliente.

Estos supuestos deben ser validados durante el proyecto para garantizar que el modelo sea preciso y eficaz.

c. Restricciones

El proyecto debe tener un presupuesto limitado: El proyecto debe ser costeable por el estudiante.

El proyecto debe tener una duración limitada: El proyecto debe completarse en un tiempo razonable.

Estas restricciones deben ser consideradas durante el desarrollo del proyecto para garantizar que el proyecto sea factible.

3.1.7. Riesgos y contingencias

a. Riesgos

El modelo puede no ser preciso: El modelo puede no ser capaz de clasificar correctamente a las personas que van a cancelar el servicio. Esto puede deberse a que la base de datos "Data Telecomunicaciones" no es fiable o a que las técnicas de minería de datos utilizadas no son eficaces.

El modelo puede no ser eficaz: El modelo puede no proporcionar información útil para predecir los clientes que van a cancelar el servicio.

El proyecto puede superar el presupuesto o la duración: El proyecto puede ser más caro o tardar más de lo previsto.

Estos riesgos pueden tener un impacto negativo en el éxito del proyecto. El equipo de proyecto debe identificar y mitigar estos riesgos para aumentar las posibilidades de éxito del proyecto.

b. Contingencias

Si el modelo no es preciso, se pueden utilizar técnicas de validación adicionales para mejorar la precisión del modelo.

Si el modelo no es eficaz, se pueden realizar cambios en el diseño del modelo para que sea más eficaz.

Si el proyecto supera el presupuesto o la duración, se pueden realizar ajustes en el plan del proyecto para reducir los costes o finalizar el proyecto a tiempo.

Las contingencias son planes de acción que se pueden implementar en caso de que se produzcan riesgos.

3.1.8. Terminología

- **Fidelización:** La fidelización de clientes es una parte muy importante en las Empresas para ganar clientes, afianzar y conservar vínculos con los clientes actuales.

- **Base de datos "Data Telecomunicaciones**": La base de datos es una base de datos sobre el comportamiento de los clientes. Cabe indicar que el *dataset* es gratuito pero las variables y parámetros son similares a la data que se maneja en la empresa.

- El *dataset* es un conjunto de parámetros de entrada y comportamiento de los clientes, se detallan datos relevantes como código único del cliente, género, si el cliente pertenece o no a la tercera edad, si es dependiente, la antigüedad que mantiene el contrato, tipo de servicio que contrató, entre otros datos fundamentales que proporcionan gran información, para a partir de éstos datos y su utilidad, analizar las mejores estrategias para evitar las cancelaciones y garantizar el crecimiento de clientes en la empresa ya que se pueden establecer métricas para mejorar la calidad y ofrecer un servicio de alta estabilidad a los clientes.

- **Minería de datos:** La minería de datos es un campo de la informática que se centra en el descubrimiento de conocimiento a partir de grandes cantidades de datos.

- En el desarrollo del proyecto se utilizará herramientas de minería de datos para desarrollar y evaluar la eficacia y precisión del modelo.

3.1.9. Objetivos de Minería de Datos

- Identificar patrones de comportamiento de clientes que preceden la cancelación de servicios en la empresa.
- Desarrollar modelos predictivos para anticipar la cancelación.
- Identificar factores clave que influyen en la retención de clientes.

Preguntas Clave:

- ¿Cuáles son los factores más influyentes en la decisión de un cliente de cancelar servicios?
- ¿Se puede prever la cancelación anticipada identificando patrones en los datos históricos?
- ¿Qué estrategias de captación de clientes han sido más exitosas en el pasado y cómo se puede mejorarlas?
- ¿Cuáles son las necesidades del cliente y que optimizaciones se han realizado después de receptar quejas de los clientes?

3.1.10. Aprendizaje automático

El aprendizaje automático es un subcampo de la minería de datos que se centra en el desarrollo de algoritmos que pueden aprender de los datos sin ser programados explícitamente. El Aprendizaje automático es fundamental para este proyecto, usando técnicas y herramientas de Inteligencia Artificial se pueden establecer modelos predictivos que ayuden a la empresa a entender como está constituida su estrategia comercial, el nivel de satisfacción que tienen los clientes actualmente en relación con el servicio, planificar mejoras y optimizaciones a implementarse para convertirse en la empresa más importante de Telecomunicaciones a nivel nacional y mundial.

- **Precisión:** La precisión de un modelo es su capacidad para clasificar correctamente a las personas que piensan abandonar la empresa, gracias a los grados de libertad y al nivel de confianza que presentan los modelos se puede tener una visión preliminar del comportamiento del cliente, gracias a la precisión de los modelos se puede tomar acción e impedir la fuga de clientes, por el contrario, afianzar la relación que mantiene con la empresa brindando un servicio de calidad.

- **Eficacia:** La eficacia de un modelo es su capacidad para proporcionar información útil para tomar medidas para reducir la deserción de los clientes en la empresa. Es muy conveniente proporcionar información importante del cliente, tener datos válidos y confiables garantiza desarrollar modelos de alta confianza, a partir de los datos observados, tomar acción e implementar métricas y puntos de mejora a nivel de servicios.

  El modelo debe ser capaz de mostrar el comportamiento del cliente, la eficacia del modelo debe proporcionar la comprensión de los niveles de satisfacción del cliente, y alertar para que se pueda tomar acción y retener a los clientes cuando no se encuentren conformes con el servicio.

- **Utilidad:** La utilidad de un modelo es su facilidad de uso y comprensión por parte de los usuarios. El modelo debe ser útil para las áreas interesadas y mostrar en tiempo real el comportamiento que están presentando los clientes. Debe ser entendible y de fácil comprensión para la audiencia, proporcionando ayuda y siendo una herramienta útil para facilitar a la empresa a la toma de decisiones.

- **Requisitos:** Los requisitos son las características que debe tener un modelo. Los requisitos son la data de los clientes, modelos de predicción, técnicas, herramientas de limpieza, normalización de la data entre otros pasos importantes durante el desarrollo.

  Una vez implementados los algoritmos debe ser requisito fundamental mostrar gráficos, imágenes, métricas, precisión del modelo y también diagramas en los que se puedan analizar de manera visual el comportamiento de los clientes.

- **Supuestos:** Los supuestos son afirmaciones que se hacen sobre la base de datos o las técnicas de minería de datos utilizadas. En los modelos de regresión, se asume que no existe una alta correlación entre las variables predictoras. La multicolinealidad puede causar problemas en la interpretación de los coeficientes del modelo.

  Se supone que los datos utilizados para entrenar y probar el modelo son representativos y válidos para el problema en cuestión. Utilizar datos no representativos puede afectar la precisión y utilidad del modelo.

- **Restricciones:** Las restricciones son limitaciones que se imponen al proyecto. Dentro de las restricciones se puede encontrar el tener datos faltantes o nulos en el

dataset, lo que involucra un proceso de limpieza, normalización, con el fin de trabajar con una data útil, que sea de gran aporte para la comprensión del estado del cliente, permita generar los modelos predictivos y visualizar los resultados de una manera fácil y práctica. Otra restricción es la falta de exactitud de los modelos, para ello se debe revisar que la data se encuentre debidamente tratada y sus variables sean consistentes.

- **Riesgos:** Los riesgos son eventos potenciales que pueden tener un impacto negativo en el proyecto. El modelo debe ser preciso y con alta fiabilidad, si el modelo no se encuentra correctamente entrenado y con datos debidamente actualizados puede producir una baja exactitud que no supere la precisión normalmente aceptable en los modelos predictivos lo que conllevaría a una mala toma de decisiones de los directivos de la empresa, incumplimientos con los resultados y objetivos por parte de las personas encargadas del proyecto y posibles llamadas de atención por no obtener resultados beneficiosos para la empresa.
  Un posible riesgo puede ser la difusión de datos sensibles de los clientes, para este caso no es un riesgo ya que la data es gratuita, sin embargo, hay que considerarlo en datos reales de los clientes.

  El mantenimiento de la data, obtención de respaldos continuos es importante para mantener los modelos actualizados y brindar una alta precisión.

- **Contingencias:** Las contingencias son planes de acción que se pueden implementar en caso de que se produzcan riesgos. En el funcionamiento el modelo puede presentar posibles fallas técnicas, se debe mantener un plan de respaldo, un modelo alternativo o una copia de seguridad, mismo que se pueda ejecutar si el modelo principal presentara fallas.

  Si los datos han sido cambiados o alterados en su construcción, pueden presentar inconsistencias al tener diferencias con la data con la que fue entrenado el modelo, en ese caso siempre se debe mantener la estructura original o establecer cambios controlados, de manera oportuna y si es posible con un historial de versiones.

  Si el modelo presenta falta de precisión es necesario volver a hacer una reingeniería con la limpieza de datos y un nuevo tratamiento con los datos actualizados.

  Mantener protocolos de seguridad para que la privacidad de los datos no sea

vulnerada, implementar seguridades en los servidores y la ejecución del modelo únicamente la debe realizar personal autorizado.

Mantenerse actualizado con las nuevas políticas de la empresa, pues si se oferta un nuevo servicio o producto se debe incorporar a la data en cuestión.

- ***Dataset*:** La data tomada como base para realizar el presente proyecto es gratuita, está basada en las características reales de los clientes de la empresa de la empresa de Telecomunicaciones, poseen valores similares en cuanto a código único, género, si el cliente tiene dependientes, si el cliente tiene pareja, si pertenece a la tercera edad, los años de antigüedad en la empresa, los servicios que posee, la forma de pago, entre otras características relevantes.

3.1.11. Costos y beneficios

a. Costos

Costos directos: Los costos directos son los gastos que se pueden atribuir directamente al proyecto. Los costos directos pueden incluir los siguientes:

Recursos humanos: Los costos de los recursos humanos incluyen los salarios, los beneficios y los gastos de viaje de los miembros del equipo del proyecto.

Las áreas de Recursos Humanos y Financiera deben estar en completo conocimiento del proyecto, ellos serán los que proporcionen los recursos económicos en lo que respecta a seguridades, pagos de horas extras que implique el desarrollo del modelo, al realizar pruebas, despliegue, entre otros.

También si implica viajar a otra sucursal por motivos de recolección de información, capacitación del modelo a Jefaturas de otras ciudades el área de Recursos Humanos debe cubrir los costos de traslado, alimentación, hospedaje y todas las comodidades para las personas que se encuentran desarrollando el proyecto.

Recursos técnicos: Los costos de los recursos técnicos incluyen los gastos de hardware, software y licencias.

Servidores robustos y con alta memoria que garantice la correcta ejecución del modelo predictivo en tiempo real y con la información completa, actualizaciones de sistema operativo, aplicaciones, antivirus actualizados y correctamente instalados. Si es que se

llegara a tener una incidencia de esta naturaleza se debe considerar servicios en la nube o adquirir servidores de alta capacidad para brindar la estabilidad a los modelos.

Se debe contar con copias de seguridad de los datos, implementar periódicamente respaldos, políticas que minimicen posibles interrupciones, perdida de conexión y comunicación con el modelo predictivo.

Contar con medidas de seguridad y planes de respaldo para minimizar el impacto de posibles fallas a nivel de infraestructura.

Recursos de datos: Los costos de los recursos de datos incluyen los gastos de recopilación, limpieza y almacenamiento.

Los datos deben ser exactos y precisos, se deben implementar protocolos para limpieza y validación de datos, así como estrategias para corregir o compensar estas deficiencias.

Ser íntegros, se deben implementar técnicas de encriptación y copias de seguridad para proteger la privacidad de los datos.

Tener estrategias para monitorear y actualizar regularmente los conjuntos de datos es fundamental para mantener la relevancia del modelo.

Costos indirectos: Los costos indirectos son los gastos que no se pueden atribuir directamente al proyecto. Los costos indirectos pueden incluir los siguientes:

Tiempos de capacitación al usuario: Una vez finalizado el proyecto se deben agendar capacitaciones con los interesados, además de la disponibilidad para brindar soporte en caso de que se presente algún problema o desconexión del modelo.

La constante limpieza, preparación de los datos, actualización son actividades que no estaban consideradas inicialmente y que son fundamentales para mantener un modelo correctamente operativo.

Costos de administración: Los costos de administración incluyen los gastos de gestión del proyecto, como la planificación, la coordinación, la total administración y control del modelo.

El tiempo dedicado por los equipos para desarrollar, probar y afinar un modelo predictivo puede ser extenso. También capacitar al personal para comprender y utilizar efectivamente el modelo puede requerir recursos considerables.

Costos en Infraestructura, licencias que respalden la seguridad del servidor donde va a estar alojado modelo predictivo.

Si el modelo presentara problemas implica gastos adicionales para la corrección y solución de posibles fallas.

Mantenimiento constante y permanente, se debe realizar un estricto monitoreo para controlar que el modelo se encuentra operando de manera eficaz.

Costos de oportunidad: Los costos de oportunidad son los beneficios que se pierden al dedicar recursos a un proyecto en lugar de a otra actividad.

Los costos de oportunidad son importantes, si el modelo funciona correctamente la oportunidad de crecimiento es significativa, por el contrario, si el modelo tiene errores, deja de funcionar o se vuelve inexacto las pérdidas de ingresos pueden ser graves.

b. Beneficios

Los beneficios se pueden dividir en dos categorías: beneficios directos e indirectos.

Beneficios directos: Son los beneficios que se pueden atribuir directamente al proyecto. Los beneficios directos pueden incluir los siguientes:

Si el modelo predictivo se ejecuta de manera correcta, puede proporcionar predicciones más precisas sobre el comportamiento del cliente, la demanda de servicios y otros aspectos claves del negocio, lo que lleva a tomar decisiones más informadas y acertadas.

Al conocer y comprender el comportamiento, necesidades y preferencias de los clientes a través de modelos predictivos, la empresa puede generar oportunidades de mejora y brindar un servicio más personalizado a cada cliente, lo que hará que el cliente se sienta conforme y atendido con el servicio y no considere cancelarlo, conduciendo a una mayor satisfacción del y su indudable fidelización.

Al tener un modelo predictivo se pueden identificar de manera oportuna los problemas y experiencia de los clientes con respecto a los servicios, lo que garantiza una rápida reacción, ejecución de soluciones y mitigar las posibles inconformidades de los clientes.

Se incrementará la competitividad en el mercado, gracias a las predicciones de los modelos, la empresa puede generar ventas cruzadas e implementar nuevos planes lo que conlleva a un importante posicionamiento en el mercado de las Telecomunicaciones.

Un modelo de fidelización de clientes preciso y eficaz: Un modelo preciso y eficaz puede ayudar a identificar a las personas que pretenden cancelar el servicio de Telecomunicaciones con la empresa y desarrollar estrategias, políticas y programas para reducir el abandono de clientes.

Información útil para tomar medidas para reducir la deserción de clientes: La información proporcionada por el modelo puede ayudar a las empresas, organizaciones públicas, privadas y a otras partes interesadas a tomar medidas para convertirse en un importante proveedor de servicios, asegurando la satisfacción del cliente.

Beneficios indirectos: Los beneficios indirectos son los beneficios que no se pueden atribuir directamente al proyecto.

Tomar mejores decisiones a través de los resultados de los modelos, de esta manera se pueden consideran planes de desarrollo a corto y largo plazo.

Generar oportunidades e introducirse en el mundo de la inteligencia artificial y en transformación digital, a través de los modelos predictivos, se está dando un paso gigante a las nuevas tendencias tecnológicas para garantizar una atención personalizada y mejorar la calidad de servicio al cliente.

Fortalecer la experiencia y fidelidad del cliente, al tener un servicio y atención de calidad el cliente se sentirá seguro y conforme, generando lealtad y fidelidad a largo plazo.

Mejora de la calidad de servicio: Un modelo preciso y eficaz puede ayudar a mejorar la calidad de servicios que recibe el cliente.

El modelo predictivo a través de una captación de intereses del cliente puede identificar oportunidades de mejora y ofertar nuevos planes. También se puede anticipar a cumplir las expectativas y necesidades del cliente.

Segmentar y clasificar de manera correcta a los clientes, en base a características similares se pueden crear planes para ciertos grupos de clientes, ofreciendo promociones innovadoras y pensadas en una mejor experiencia del cliente.

A través de un modelo predictivo se pueden identificar los servicios más usados, los que más problema generan y en base a esos indicadores implementar

desarrollos y optimizaciones para estabilizar el servicio que presenta más inconformidades.

Reducción del abandono de clientes: Un modelo preciso y eficaz puede ayudar a reducir la deserción de clientes de la empresa de Telecomunicaciones.

Comprender de mejor manera las necesidades y preferencias de los clientes se pueden implementar estrategias y campañas de retención, ofrecer mayor velocidad, bajar de precio un plan o brindar productos para premiar la fidelidad de los clientes.

El modelo puede identificar el comportamiento del cliente a través de sus datos históricos y de esta manera intervenir de forma rápida y proactiva para retener al cliente.

Mejorar la atención de servicio al cliente, en base a las quejas de los clientes se puede tomar decisiones con respecto al contacto con el cliente, capacitar al personal, incluso los jefes superiores pueden llamarlos para que el cliente se sienta atendido e importante.

Opiniones de usuarios, a través de encuestas se puede tener un *feedback* de los clientes y considerar los puntos de mejora.

3.1.12. Evaluación inicial de herramientas y técnicas.

A continuación, se presentan algunas consideraciones para la evaluación de herramientas y técnicas para el problema en estudio:

Precisión: Las herramientas y técnicas deben ser capaces de clasificar correctamente a las personas que pretenden cancelar los servicios con la empresa de Telecomunicaciones.

La precisión es la proporción de observaciones positivas correctamente predichas con respecto al total de positivos predichos. Es una medida de cuántas de las instancias positivas predichas son realmente positivas.

Eficacia: La información proporcionada por el modelo puede ayudar a las empresas, organizaciones públicas y privadas y a otras partes interesadas a tomar medidas para garantizar ser un buen proveedor de servicios, asegurando la satisfacción del cliente.

Un modelo eficaz produce predicciones precisas y confiables. Cuanto más cerca estén las predicciones de la realidad, mayor será la eficacia del modelo.

Utilidad: Las herramientas y técnicas deben ser fáciles de usar y comprender

por los usuarios.

Los modelos predictivos permiten prever eventos futuros o comportamientos basados en datos históricos. En telecomunicaciones, esto puede significar anticipar la demanda de servicios, la rotación de clientes o incluso problemas en la red antes de que ocurran.

Los modelos ayudan a comprender mejor las necesidades y preferencias de los clientes, permitiendo ofrecer servicios más personalizados y adaptados a las necesidades individuales, lo que mejora la satisfacción y fidelización del cliente.

La evaluación de herramientas y técnicas es una parte importante del proceso de desarrollo de un modelo de minería de datos. La evaluación adecuada puede ayudar a garantizar que se utilicen las herramientas y técnicas adecuadas para el problema en cuestión.

3.2. Entendimiento de los datos

El análisis inicial de datos en el proceso CRISP-DM es fundamental y esencial para entender la estructura y naturaleza de los datos que se utilizarán en un proyecto de minería de datos. Esta etapa permite comprender la calidad, la forma y la estructura de los datos antes de sumergirse en análisis más detallados.

La fase de entendimiento de los datos inicia con la recolección de datos y la familiarización con los mismos. Identifica y analiza problemas en la calidad de los datos para aplicar técnicas en su etapa de preparación. Ayuda a conocer qué herramientas y enfoques serán más efectivos para el trabajo de implementación y construcción de modelos predictivos.

3.2.1. Recopilación de Datos Iniciales

- **Informes mensuales de cancelaciones:** Detalles sobre los clientes que cancelaron servicios, incluyendo razones (si se proporcionaron) y fechas de cancelación.

- **Encuestas de satisfacción del cliente:** Retroalimentación directa de los clientes sobre su experiencia y motivos de satisfacción o insatisfacción.

- **Datos de uso de servicios:** Información sobre la utilización de servicios específicos por parte de los clientes, como servicios de Internet, TV, *Streaming*, líneas telefónicas, etc.

- **Datos de facturación:** Historial detallado de facturación de los clientes, incluyendo cargos mensuales, descuentos y cualquier información relacionada con pagos.

- **Datos demográficos del cliente:** Información sobre género, edad y cualquier otro dato demográfico que aporte al modelo a determinar un comportamiento relevante para incluir en el desarrollo.

- **Datos de retención de clientes anteriores:** Información sobre clientes que cancelaron el servicio en el pasado y las circunstancias que influyeron en la cancelación. En base a ellos se puede inferir en el comportamiento de los clientes actuales que pretenden cancelar el servicio.

3.2.1.1. Métodos de recopilación

- **Sistemas Internos de la empresa:** Acceso a la base de datos interna de la empresa para extraer información detallada sobre los clientes en los periodos deseados.

- **Encuestas actuales y entrevistas:** Realizar encuestas adicionales y/o entrevistas con clientes actuales y con aquellos que cancelaron el servicio para obtener información cualitativa adicional y determinar los motivos de cancelación, de esta información se analizarán puntos de mejora para los clientes actuales.

- **Colaboración con departamentos internos:** Colaboración con departamentos y áreas de atención al cliente, marketing, departamento financiero, tecnología y ventas para obtener información adicional sobre interacciones con los clientes y sus expectativas del servicio.

- La fase de entendimiento de los datos inicia con la recolección y la familiarización de los datos. Identifica problemas de la calidad de los datos. Posteriormente los datos se convertirán en información.

Se revisa inicialmente la data “Data Telecomunicaciones.csv” completa, se seleccionarán las variables que llegan a ser útiles y relevantes, se realiza una exploración de los campos seleccionados con el fin de determinar con qué tipo de datos se está trabajando, hay diferente tipo de datos como son valores numéricos, datos booleanos, cadenas de caracteres, fechas, entre otras. Una vez entendida y definida la data se asegura

que se va a trabajar con datos íntegros y que aporten información importante para su tratamiento, análisis y ejecución.

a. Recopilar datos iniciales

Los datos iniciales se han proporcionado de una data que describe información sobre una lista de clientes de una empresa de servicios de Telecomunicaciones. Para ello se han determinado ciertos indicadores que denotan la posible cancelación de un servicio.

Es importante conocer los parámetros de entrada e información relevante, ya que a partir de ellos se pueden realizar acciones para evitar la fuga de clientes y al contrario ganar más.

b. Exploración inicial de los datos

Para tener una data de buena calidad y que permita comprender de manera clara y exacta el comportamiento de los clientes se realizará el siguiente proceso.

- **Revisión de estructura de datos:** Examinar la estructura y el formato de los conjuntos de datos para comprender las variables y su organización.
- **Análisis estadístico preliminar:** Realizar análisis estadísticos básicos para obtener una visión general de las distribuciones, tendencias y posibles correlaciones en los datos.
- **Identificación de valores atípicos, nulos o vacíos:** Buscar valores atípicos o atípicos que puedan afectar la calidad de los datos.
- **Exploración de relaciones:** Identificar posibles relaciones entre las variables, especialmente aquellas relacionadas con la cancelación de servicios.
- **Visualización de datos**: Utilizar gráficos y visualizaciones para representar de manera efectiva la información y patrones presentes en los datos.

c. Informe inicial de recopilación de datos

Comprender la Naturaleza de los Datos

- Identificar la tipología de las variables (categóricas, numéricas, binarias).
- Evaluar la distribución y la variabilidad de los datos.
- Reconocer la presencia de valores faltantes o atípicos.

Familiarizarse con las variables

- Describir cada variable en términos de su significado y relevancia para el negocio.
- Entender las relaciones potenciales entre las variables.

Verificar la calidad de los datos

- Realizar una limpieza inicial de datos, abordando valores nulos o inconsistentes.
- Evaluar la coherencia de los datos con respecto al contexto del negocio.

Identificar patrones preliminares:

- Explorar visualmente los datos para identificar patrones, tendencias o posibles correlaciones.
- Realizar análisis estadísticos básicos para obtener *insights* iniciales.

3.2.1.2. Análisis del dataset "Data Telecomunicaciones.csv"

En el data set Data Telecomunicaciones.csv se tiene un tamaño de 7043 registros en los que se detallan campos como el género, antigüedad, servicio telefónico, internet, streaming, entre otras que se describirán más adelante.

En la **Figura 3** se observa la carga del dataset a Python, en este proyecto se trabajará con Google Collab.

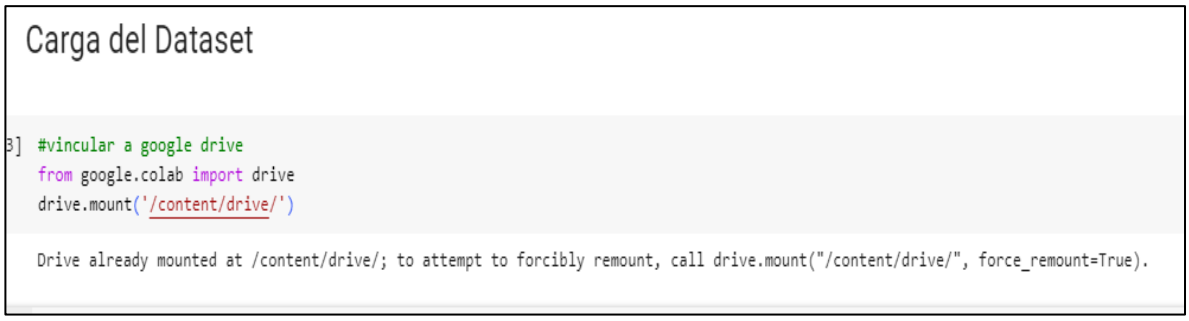


Figura 3: Vincular a Google drive. Autor: Blanca Mejía, 2024.

El archivo "Data Telecomunicaciones.csv" previamente fue almacenado en la carpeta CIENCIA DE DATOS en My Drive de la cuenta de Google con el propósito que el archivo se encuentre siempre disponible en el directorio y no realizar la carga cada vez que se conecte a Google Collab.

En la última parte de la **Figura 4** se observa que el *dataset* contiene 23

columnas.

```
#ruta = r'C:\Users\HP\Downloads\WA_Fn-UseC_-Telco-Customer-Churn.csv'
ruta = "/content/drive/My Drive/CIENCIA DE DATOS/Data Telecomunicaciones.csv"
Clientes = pd.read_csv(ruta)
Clientes.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	S
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	

5 rows × 23 columns

Figura 4: Carga de dataset "Data Telecomunicaciones". Autor: Blanca Mejía, 2024.

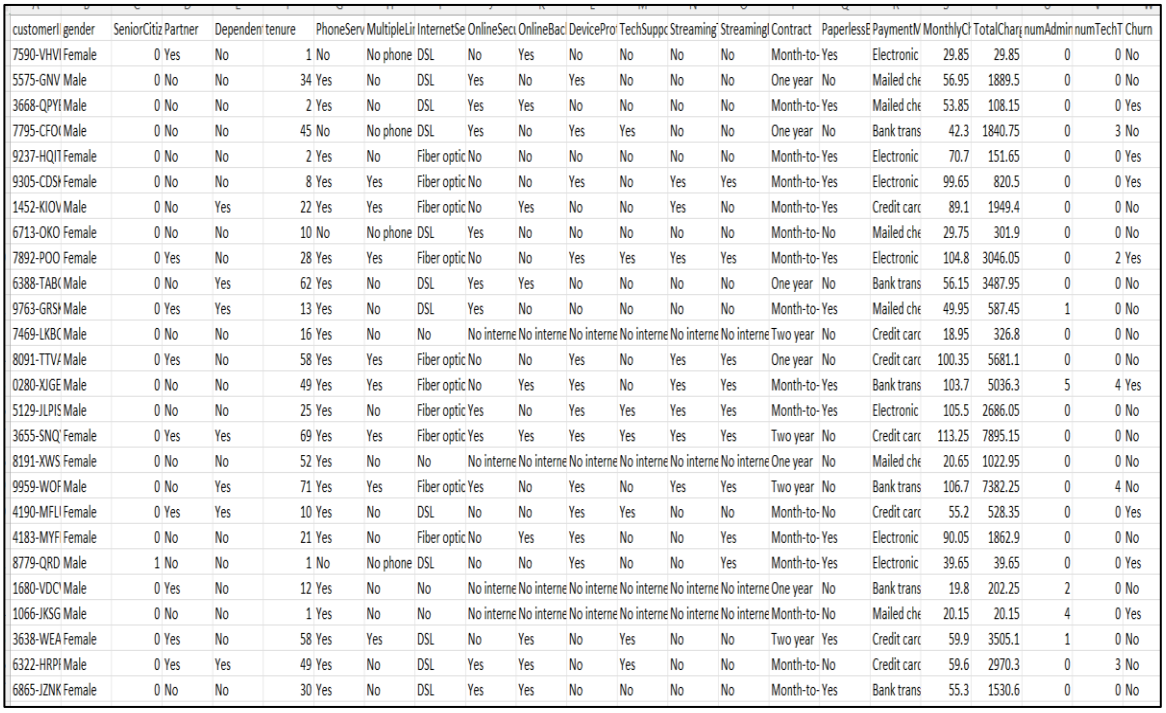
Cantidad de registros del *dataset* “Data Telecomunicaciones.csv”. En **la Figura 5** se pueden apreciar la cantidad de 7043 registros.

```
[8]  len(Clientes)

     7043
```

Figura 5: Cantidad de registros dataset. Autor: Blanca Mejía, 2024

a. Describir datos

En esta fase se realizará una breve descripción e informe de descripción de los datos con su significado y el papel que cada dato representa en el *dataset*. En esta fase se realiza el detalle de todas las variables recolectadas, se realizan análisis estadísticos básicos y visualizaciones para comprender el comportamiento de cada dato y el valor que aporta al modelo. En la **Figura 6** se previsualiza la data desde Excel.



customerI	gender	SeniorCitiz	Partner	Dependen	tenure	PhoneServ	MultipleLi	InternetSe	OnlineSec	OnlineBac	DevicePro	TechSuppc	Streaming	Streaming	Contract	PaperlessE	PaymentM	MonthlyCl	TotalChar	numAdmin	numTechT	Churn
7590-VHVI	Female	0	Yes	No	1	No	No phone	DSL	No	Yes	No	No	No	No	Month-to-	Yes	Electronic	29.85	29.85	0	0	No
5575-GNV	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	No	No	No	One year	No	Mailed che	56.95	1889.5	0	0	No
3668-QPY	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-	Yes	Mailed che	53.85	108.15	0	0	Yes
7795-CFO	Male	0	No	No	45	No	No phone	DSL	Yes	No	Yes	Yes	No	No	One year	No	Bank trans	42.3	1840.75	0	3	No
9237-HQIT	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No	Month-to-	Yes	Electronic	70.7	151.65	0	0	Yes
9305-CDS	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes	Month-to-	Yes	Electronic	99.65	820.5	0	0	Yes
1452-KIOV	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No	No	Yes	No	Month-to-	Yes	Credit carc	89.1	1949.4	0	0	No
6713-OKO	Female	0	No	No	10	No	No phone	DSL	Yes	No	No	No	No	No	Month-to-	No	Mailed che	29.75	301.9	0	0	No
7892-POO	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	No	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	104.8	3046.05	0	2	Yes
6388-TABG	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No	No	No	No	One year	No	Bank trans	56.15	3487.95	0	0	No
9763-GRS	Male	0	Yes	Yes	13	Yes	No	DSL	Yes	No	No	No	No	No	Month-to-	Yes	Mailed che	49.95	587.45	1	0	No
7469-LKBC	Male	0	No	No	16	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	Two year	No	Credit carc	18.95	326.8	0	0	No
8091-TTVA	Male	0	Yes	No	58	Yes	Yes	Fiber optic	No	No	Yes	No	Yes	Yes	One year	No	Credit carc	100.35	5681.1	0	0	No
0280-XJGE	Male	0	No	No	49	Yes	Yes	Fiber optic	No	Yes	Yes	No	Yes	Yes	Month-to-	Yes	Bank trans	103.7	5036.3	5	4	Yes
5129-JLPIS	Male	0	No	No	25	Yes	No	Fiber optic	Yes	No	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	105.5	2686.05	0	0	No
3655-SNQ	Female	0	Yes	Yes	69	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	No	Credit carc	113.25	7895.15	0	0	No
8191-XWS	Female	0	No	No	52	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	One year	No	Mailed che	20.65	1022.95	0	0	No
9959-WOF	Male	0	No	Yes	71	Yes	Yes	Fiber optic	Yes	No	Yes	No	Yes	Yes	Two year	No	Bank trans	106.7	7382.25	0	4	No
4190-MFLI	Female	0	Yes	Yes	10	Yes	No	DSL	No	No	Yes	Yes	No	No	Month-to-	No	Credit carc	55.2	528.35	0	0	Yes
4183-MYF	Female	0	No	No	21	Yes	No	Fiber optic	No	Yes	Yes	No	No	Yes	Month-to-	Yes	Electronic	90.05	1862.9	0	0	No
8779-QRD	Male	1	No	No	1	No	No phone	DSL	No	No	Yes	No	No	Yes	Month-to-	Yes	Electronic	39.65	39.65	0	0	Yes
1680-VDC	Male	0	Yes	No	12	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	One year	No	Bank trans	19.8	202.25	2	0	No
1066-JKSG	Male	0	No	No	1	Yes	No	No	No interne	No interne	No interne	No interne	No interne	No interne	Month-to-	No	Mailed che	20.15	20.15	4	0	Yes
3638-WEA	Female	0	Yes	No	58	Yes	Yes	DSL	No	Yes	No	Yes	No	No	Two year	Yes	Credit carc	59.9	3505.1	1	0	No
6322-HRP	Male	0	Yes	Yes	49	Yes	No	DSL	Yes	Yes	No	Yes	No	No	Month-to-	No	Credit carc	59.6	2970.3	0	3	No
6865-JZNK	Female	0	No	No	30	Yes	No	DSL	Yes	Yes	No	No	No	No	Month-to-	Yes	Bank trans	55.3	1530.6	0	0	No

Figura 6: Registros observados desde Excel. Autor: Blanca Mejía, 2024.

Resumen de variables y datos: A través de comandos y funciones ya implementadas en Python se realizan cálculos para obtener la media, moda, desviación estándar, mínimos, máximos, entre otros cálculos útiles para empezar a tener un panorama claro de las variables.

Visualizaciones y gráficos: Se realizan gráficos como *boxplot*, diagrama de barras, caja de bigotes, dispersión, histogramas, pasteles, entre otros.

Identifica *outliers*: Permite visualizar *outliers*, valores fuera del rango normal, valores anómalos, vacíos, redundantes y todo tipo de datos incongruentes que afecten a la data y a su correcto análisis.

b. Informe de descripción de datos

Para realizar el informe se procederá a identificar las variables a usarse, a continuación, se describen las mismas.

En la **Tabla 1** se realiza el detalle y descripción de cada variable de las cuales está compuesto el conjunto de datos.

TABLA DE DESCRIPCIÓN DE DATOS

1	customerID	Representa el número único y clave del cliente en el conjunto de datos.
2	gender	Género del cliente, Male (Hombre), Female (Mujer).
3	SeniorCitizen	Indica si el cliente es de la tercera edad. 1 (Si), 0 (No).
4	Partner	Indica si el cliente tiene pareja. Yes (Si), No (No).
5	Dependents	Indica si el cliente tiene dependientes a su responsabilidad. Yes (Si), No (No).
6	tenure	Antigüedad que tiene el cliente en la empresa, número de meses.
7	PhoneService	Indica si el cliente posee servicio telefónico. Yes (Sí), No (No).
8	MultipleLines	Indica si el cliente posee múltiples líneas telefónicas. Yes (Sí), No (No), No phone service (Sin servicio telefónico).
9	InternetService	Indica si el cliente tiene servicio de Internet. DSL, Fiber optic, No.
10	OnlineSecurity	Indica si el cliente se ha registrado para el servicio de seguridad en línea. Yes (Sí), No (No), No internet service (Sin servicio de internet).
11	OnlineBackup	Indica si es cliente se ha registrado para recibir una copia en línea. Yes (Sí), No (No), No internet service (Sin servicio de internet).
12	DeviceProtection	Indica si el cliente se ha registrado para la protección del dispositivo. Yes (Sí), No (No), No internet service (Sin servicio de internet).
13	TechSupport	Indica si el cliente se ha registrado para recibir soporte técnico. Yes (Sí), No (No), No internet service (Sin servicio de internet).
14	StreamingTV	Indica si el cliente se ha registrado para recibir transmisión de TV. Yes (Sí), No (No), No internet service (Sin servicio de internet).
15	StreamingMovies	Indica si el cliente se ha registrado para recibir transmisión de películas. Yes (Sí), No (No), No internet service (Sin servicio de internet).
16	Contract	Indica la duración del cliente en la empresa. Month-to-month, One year, Two Year.
17	PaperlessBilling	Indica si el cliente posee facturación electrónica. Yes (Sí), No (No).
18	PaymentMethod	Indica el tipo y método de pago del cliente. Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic).
19	MonthlyCharges	Indica el cargo mensual que ha realizado el cliente.
20	TotalCharges	Indica el cargo total de consumo del cliente actualmente.
21	numAdminTickets	Indica el número de *tickets* que tiene abierto el cliente a la empresa.
22	numTechTickets	Indica el número de *tickets* técnicos que tiene abierto el cliente a la empresa.
23	Churn	Indica si el cliente ha cancelado el servicio de la empresa o no. Yes (Sí), No (No).

Tabla 1: Descripción del conjunto de datos. Autor: Blanca Mejía, 2024.

c. Explorar los Datos

El análisis inicial del conjunto de datos revela la información, estructura, tipo y naturaleza que tiene cada variable. En la **Figura 7** se observa que el conjunto de datos tiene una cantidad de 7043 registros y 23 variables entre numéricas (*int64*), flotante (*float64*) y categóricas (*object*).

```
Clientes.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 23 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  numAdminTickets   7043 non-null   int64
 21  numTechTickets    7043 non-null   int64
 22  Churn             7043 non-null   object
dtypes: float64(1), int64(4), object(18)
memory usage: 1.2+ MB
```

Figura 7: Exploración de datos en Python. Autor: Blanca Mejía, 2024.

Dentro de la data se puede identificar lo siguiente:

- La columna *MonthlyCharges* es de tipo float64.
- Las columnas *tenure*, *numAdminTickets* y *numTechTickets* son de tipo int64.
- Las variables restantes son tipo *object*, lo que indican que son categóricas.
- En el resumen se puede observar que no existen valores nulos, lo que proporciona alta confianza en la data y evita realizar procesos de limpieza y posible eliminación de datos. Al tener datos *non-null* garantiza la integridad de los datos y facilita la continuación del proceso ayudando a comprender de mejor manera la relevancia de cada variable.

Variables numéricas:

En del dataset se observan las siguientes variables numéricas: *SeniorCitizen, tenure, numAdminTickets, numTechTickets.*

En la **Figura 8** se realiza un describe en las variables numéricas para obtener la estadística descriptiva, como media, desviación estándar y cuartiles.

```
#Estadísticas de variables numéricas
Clientes.describe()
```

	SeniorCitizen	tenure	MonthlyCharges	numAdminTickets	numTechTickets
count	7043.000000	7043.000000	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692	0.515689	0.419566
std	0.368612	24.559481	30.090047	1.275299	1.250117
min	0.000000	0.000000	18.250000	0.000000	0.000000
25%	0.000000	9.000000	35.500000	0.000000	0.000000
50%	0.000000	29.000000	70.350000	0.000000	0.000000
75%	0.000000	55.000000	89.850000	0.000000	0.000000
max	1.000000	72.000000	118.750000	5.000000	9.000000

Figura 8: Estadísticas descriptivas variables numéricas. Autor: Blanca Mejía, 2024.

- El análisis de las variables numéricas revela la valiosa representación de cada variable, por ejemplo, en demografía, y el comportamiento de los clientes de la empresa.

- La variable *SeniorCitizen* indica que el 16% de los clientes son de la tercera edad.

- La variable *tenure* indica el tiempo que el cliente tiene contratado el servicio, tiene una media de 32 meses. Los clientes con 0 meses indica que varios clientes han contratado recientemente el servicio. También se tiene un máximo de 72 meses, lo que es bueno para la empresa, está indicando que hay una fidelidad presente a largo plazo.

- La variable *MonthlyCharges* indica una media de 64%, indica que hay variedad en las tarifas de planes de la empresa.

- En cuanto a *numAdminTickets* y *numTechTickets* indican la presencia de requerimientos en la mesa de servicios y *call center* de la empresa, con una media de 0.51 y 0.41 respectivamente.

- Para las variables categóricas, se propone emplear describe(include='O') para obtener estadísticas descriptivas tales *como count, unique, top, freq* y estadísticas específicas de este tipo de datos.

Variables no numéricas o categóricas:

En del dataset se observan las siguientes variables no numéricas o categóricas customerID, gender, Partner, Dependents, PhoneService, MultipleLines, Inernet, Service, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, Streaming TV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod, TotalCharges, Churn.

En la **Figura 9** se realiza un describe en las variables no numéricas o categóricas para obtener las estadísticas descriptivas.

```
# ANALISIS EXPLORATORIO
# DESCRIPCION GENERAL DE LOS DATOS CATEGORICOS
descripcion_categorica = Clientes.describe(include='O')
print(descripcion_categorica)
```

```
        customerID gender Partner Dependents PhoneService MultipleLines  \
count         7043   7043    7043       7043         7043          7043
unique        7043      2       2          2            2             3
top     7590-VHVEG   Male      No         No          Yes            No
freq             1   3555    3641       4933         6361          3390

       InternetService OnlineSecurity OnlineBackup DeviceProtection  \
count             7043           7043         7043             7043
unique               3              3            3                3
top        Fiber optic             No           No               No
freq              3096           3498         3088             3095

       TechSupport StreamingTV StreamingMovies        Contract  \
count         7043        7043            7043            7043
unique           3           3               3               3
top             No          No              No  Month-to-month
freq          3473        2810            2785            3875

       PaperlessBilling     PaymentMethod TotalCharges Churn
count              7043              7043         7043  7043
unique                2                 4         6531     2
top                 Yes  Electronic check                 No
freq               4171              2365           11  5174
```

Figura 9: Estadísticas descriptivas variables categóricas. Autor: Blanca Mejía, 2024.

En este análisis se va a revisar las variables categóricas y la visión detallada que cada una representa en el conjunto de datos. Se observa que la columna *customerID* posee 7043 registros únicos, lo que indica que cada cliente tiene un identificador único, lo cual es bueno porque evita quitar datos duplicados en la preparación de la data.

Las variables *gender, Partner, Dependents, PhoneService* y *PaperlessBilling* son predominantemente binarias, con dos categorías posibles. La variable *gender* muestra una distribución casi equitativa entre *male* y *female*, mientras que *Partner* y *Dependents* tienen más frecuencias de No en comparación con Yes. *PhoneService* y *PaperlessBilling*, por otro lado, muestran una predominancia de Yes.

En lo que respecta a las variables vinculadas a los servicios, como *MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV* y *StreamingMovies*, tienen más de una opción. Por ejemplo, *MultipleLines* tiene tres posibilidades: No, *Yes* y *No phone service*, siendo esta última la menos común. La variable *Contract*, que describe el tipo de contrato, muestra una clara preferencia por contratos de corto plazo, destacando una proporción significativa de clientes que optan por contratos mensuales.

En cuanto a *PaymentMethod*, presenta cuatro opciones, siendo *Electronic check* la más utilizada. Además, la variable *Churn*, que indica si un cliente ha cancelado el servicio, muestra una frecuencia alta de "No", sugiriendo que la mayoría de los clientes en el conjunto de datos no han cancelado el servicio.

GRÁFICOS ESTADÍSTICOS

Se obtienen gráficos estadísticos del conjunto de datos utilizado para el proyecto, se pueden evidenciar los siguientes resultados. En la **Figura 10** se observa que el 73.46% de clientes no abandonaron o cancelaron el servicio, mientras que un 26.54% si cancelaron el servicio.

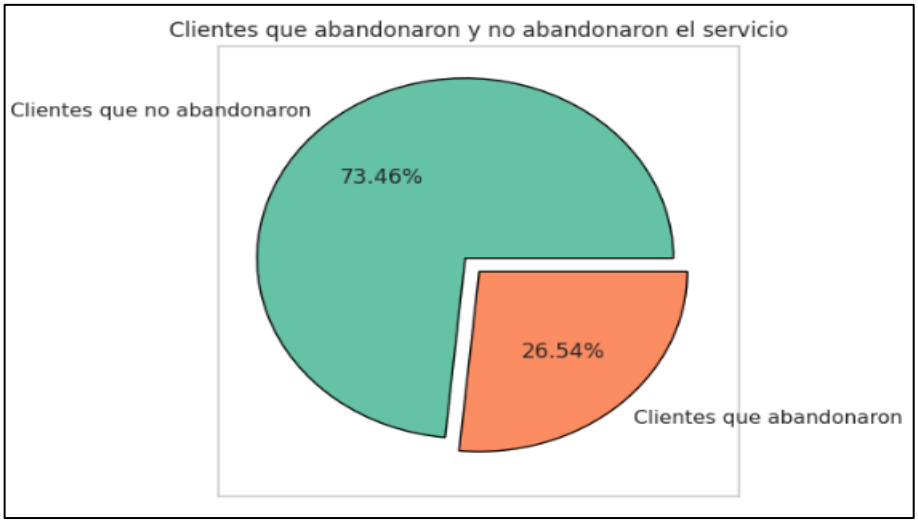


Figura 10. Clientes que abandonaron y no abandonaron. Autor: Blanca Mejía, 2024.

En la **Figura 11** se observa que 5174 de clientes no abandonaron o cancelaron el servicio, mientras que 1869 si cancelaron el servicio.

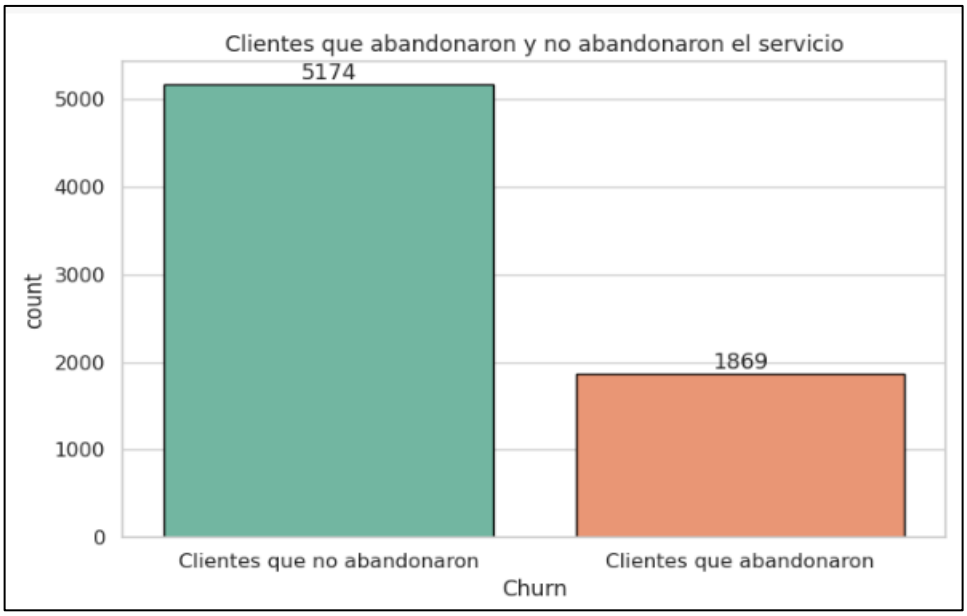


Figura 11: Clientes que abandonaron y no abandonaron. Autor: Blanca Mejía, 2024.

Análisis estadístico con la columna ***gender***

En la **Figura 12** se observa que el 3488 de clientes son de género femenino, mientras que 3555 son de género masculino, el conjunto de datos esta casi equilibrado.

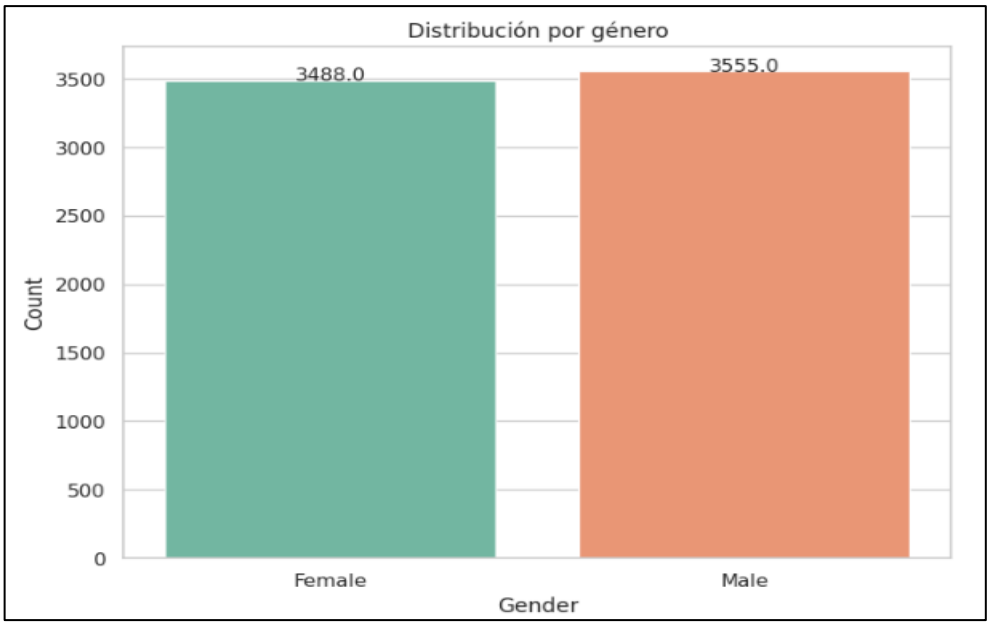


Figura 12: Cantidad de clientes por género. Autor: Blanca Mejía, 2024.

En la **Figura 13** se observa que un total de 2549 de género femenino no cancelaron el servicio, mientras que 939 si cancelaron el servicio. En la **Figura 13** se observa que un total de 2625 de género masculino no cancelaron el servicio, mientras que 930 si cancelaron el servicio.

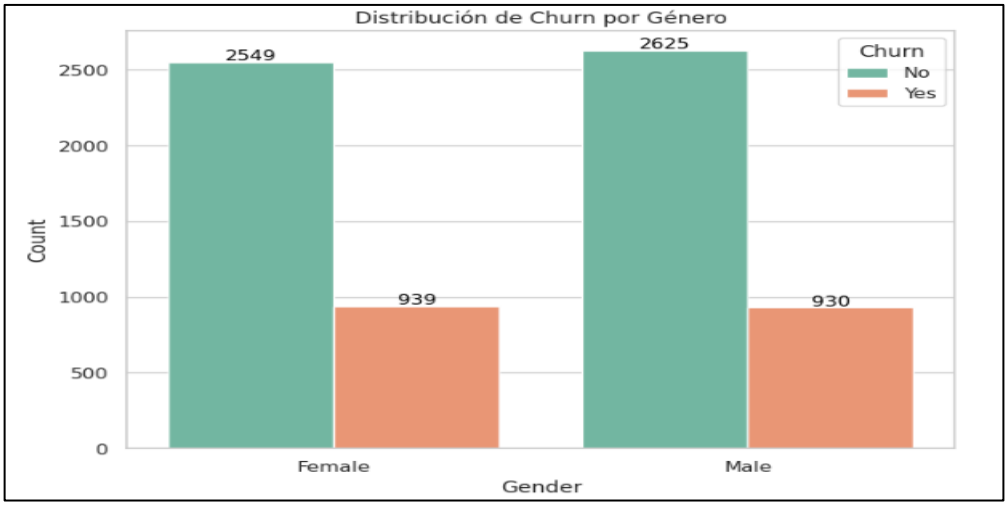


Figura 13: Cantidad de clientes por género. Autor: Blanca Mejía, 2024.

Análisis estadístico con la columna ***SeniorCitizen***

En la **Figura 14** se observa que el 5901 de clientes no son de tercera edad, mientras que 1142 si lo son.

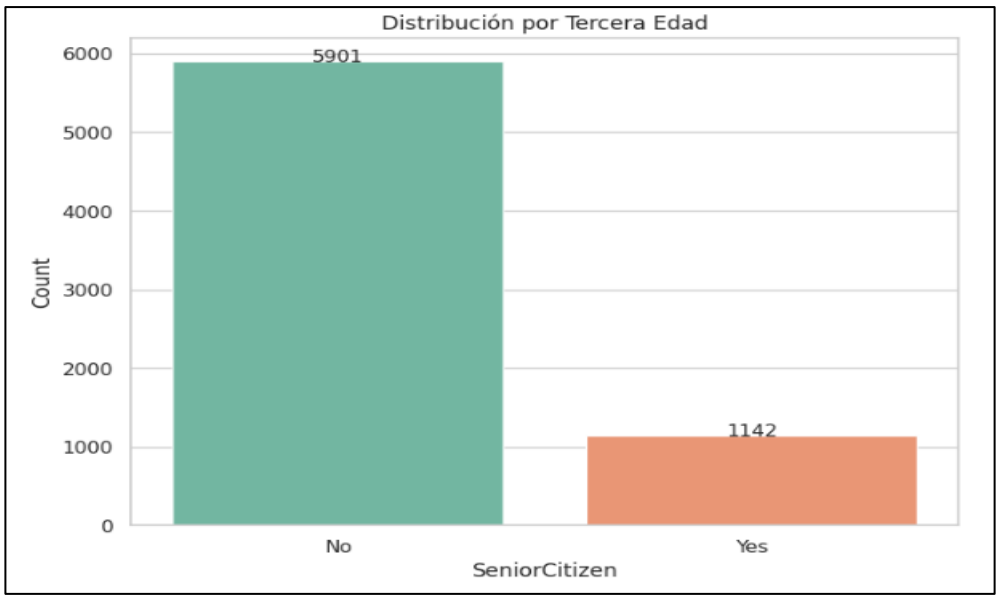


Figura 14: Cantidad de clientes tercera edad. Autor: Blanca Mejía, 2024.

En la **Figura 15** se observa que un total de 4508 de tercera edad no cancelaron el servicio, mientras que 1393 si cancelaron el servicio. En la **Figura 15** se observa que un total de 666 de tercera edad no cancelaron el servicio, mientras que 476 si cancelaron el servicio.

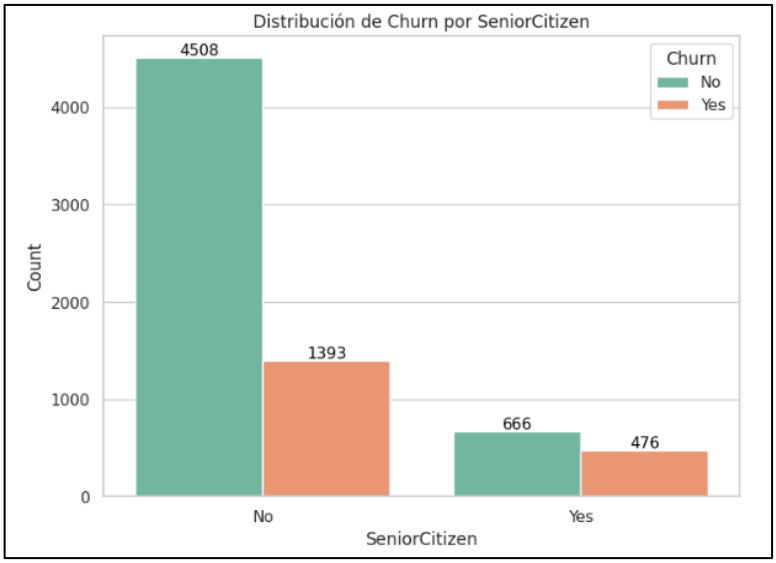


Figura 15: Cantidad de clientes tercera edad. Autor: Blanca Mejía, 2024.

Análisis estadístico con la columna ***Partner***

En la **Figura 16** se observa que el 3402 de clientes si tienen pareja, mientras que 3641 no tienen pareja.

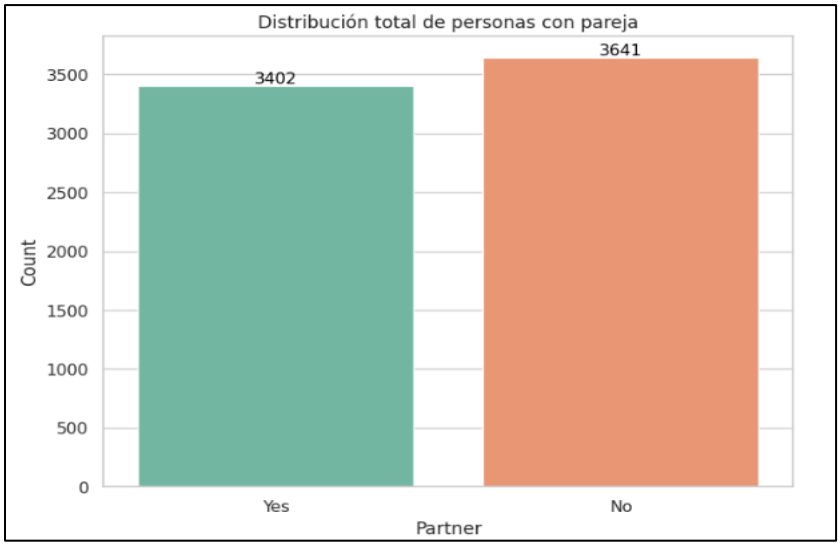


Figura 16: Cantidad de clientes que tienen pareja. Autor: Blanca Mejía, 2024.

En la **Figura 17** se observa que un total de 2733 de clientes que tienen pareja no cancelaron el servicio, mientras que 669 de clientes que tienen pareja si cancelaron el servicio. En la **Figura 17** se observa que un total de 2441 de clientes que no tienen pareja no cancelaron el servicio, mientras que 1200 de clientes que no tienen pareja si cancelaron el servicio.

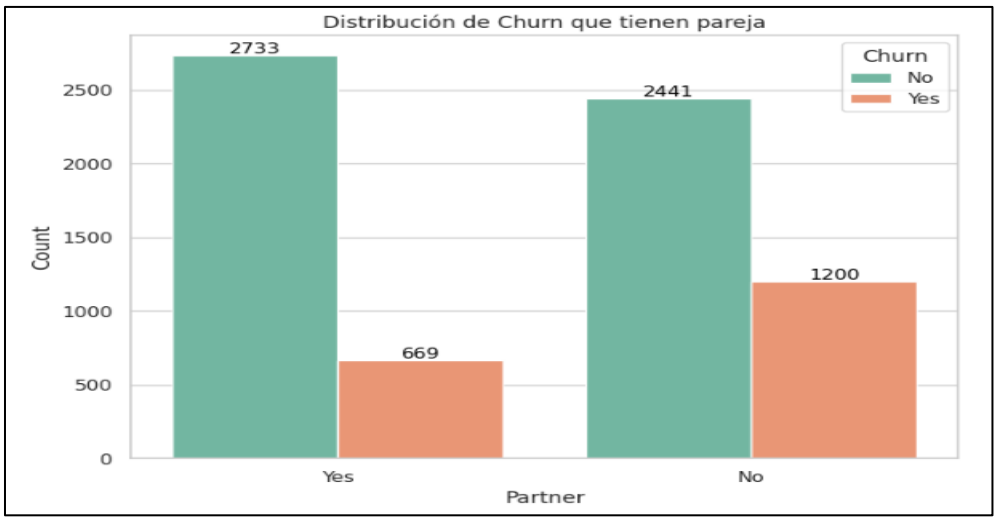


Figura 17: Clientes con pareja y que cancelaron el servicio. Autor: Blanca Mejía, 2024.

Verificar la calidad de los datos:

La verificación de la calidad de los datos es muy importante, aquí se determina que los datos sean confiables y aptos para un buen modelado y un nivel de exactitud.

Revisar tamaño del Dataset:

En la **Figura 18** se puede observar que el *dataset* consta de 7043 registros, con 23 columnas.

```
Clientes.shape

(7043, 23)
```

Figura 18: Tamaño del Dataset. Autor: Blanca Mejía, 2024.

Datos nulos:

En la **Figura 19** se puede apreciar que no existen datos nulos, los datos están limpios y completos, sin presentar valores nulos en ninguna columna. Esta revisión es fundamental para garantizar la integridad y evita hacer ajustes futuros, asegura que las

variables numéricas y categóricas son completas para cada cliente. Al tener 0 valores nulos indican un conjunto de datos muy confiable y sólido.

```
#REVISO VALORES NULOS
Clientes.isnull().sum()
```

```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
numAdminTickets     0
numTechTickets      0
Churn               0
dtype: int64
```

Figura 19: Datos nulos del dataset. Autor: Blanca Mejía, 2024.

Esta revisión es alentadora para continuar con la implementación del modelo predictivo ya que se asegura que la data a tratar se encuentra completa y garantiza que se puede para proceder con análisis más profundos y la aplicación de técnicas de minería de datos.

Verificar la calidad de los datos:

En términos generales, la calidad de los datos es bastante buena, el total de valores no existen valores nulos y no se encuentran errores en el dataset. Posterior a este análisis se considera que existe suficiente data de calidad para cumplir con los objetivos planteados en las etapas anteriores.

3.3. Preparación de los datos

Cubre todas las actividades necesarias para construir el *dataset* final, las tareas incluyen tablas, registros, selección de atributos, así como transformación y limpieza de los datos para las herramientas de modelado.

3.3.1. Preparación de la data

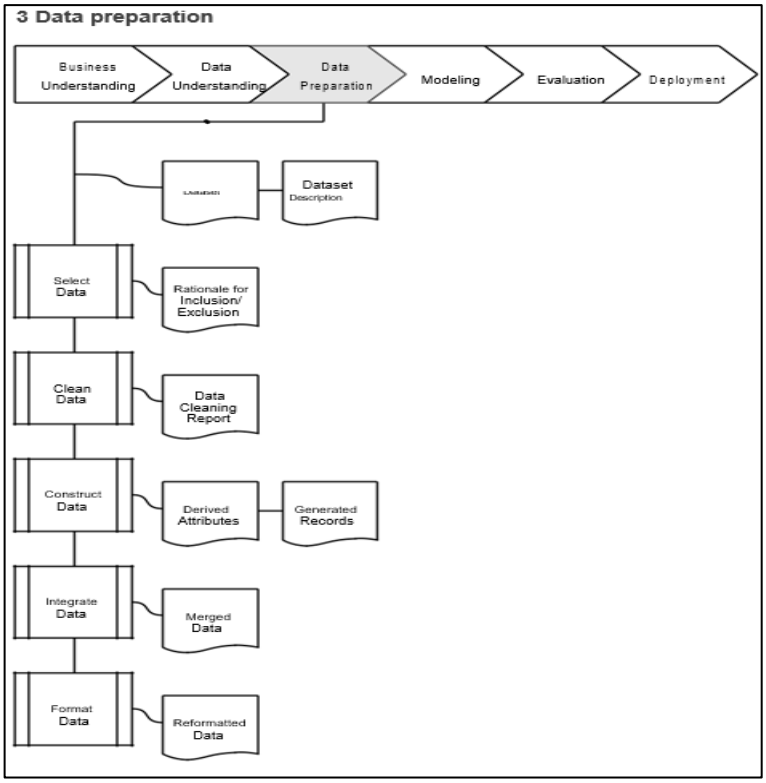


Figura 20: Fase Data Preparation. Autor: Desconocido, recuperado 2024.

En la **Figura 20** se observa la fase de *Data Preparation*, esta fase es por lo general la más extensa y laboriosa en un proyecto de modelos predictivos, se deben revisar que la data cumpla todos los estándares, la parte crucial es seleccionar las variables con las que se trabajará y la importancia que cada una de ellas representa y aporta al modelo.

A partir de un *dataset* crudo y con datos en bruto se determina la selección de datos, la limpieza, transformación, normalización, entre otros procedimientos. El dataset contiene datos relacionados con una empresa de Telecomunicaciones, la información se centra en los clientes y sus atributos personales (como género, estado civil, edad, entre otros), sus detalles vinculados a los servicios actualmente con la empresa, (tipo de contrato, servicios suscritos, método de pago, etc.) y la variable objetivo, que es la variable *'Churn'*, indicando si el cliente ha cancelado o no el servicio con la empresa.

3.3.1.1. Seleccionar datos (relacionar para inclusión / exclusión)

Se ha decidió eliminar la variable *customerID* ya que la información no es relevante, es únicamente el código de cada cliente, pero al ser distinto para cada cliente no

es de mayor aporte en este proyecto. Como se observa en la **Figura 21** la variable *customerID* muestra información irrelevante, por esa razón se eliminará más adelante.

```
#Visualización del conjunto de datos
Clientes.head(10)
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No
5	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No

Figura 21: Visualización registros del dataset. Autor: Blanca Mejía, 2024.

Se trabajará con las 22 variables restantes, cada variable proporciona información importante del comportamiento de un cliente con respecto a la variable objetivo "*Churn*", la completitud e integridad de la data también ha ayudado a tomar esta decisión.

3.3.1.2. Limpieza de la data

Eliminar valores nulos: Como se observa en la **Figura 22**, la variable *TotalCharges* tiene 11 registros nulos, por esa razón se ha decidido reemplazar con la mediada de los datos de la misma columna con el objetivo de tener valores completos.

```
Clientes[Clientes['TotalCharges']==' ']
```

rCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	numAdminTickets	numTechTickets	Churn
0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Yes	No	Two year	Yes	Bank transfer (automatic)	52.55		0	0	No
0	No	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service	Two year	No	Mailed check	20.25		5	0	No
0	Yes	Yes	0	Yes	No	DSL	Yes	...	Yes	Yes	Two year	No	Mailed check	80.85		0	0	No
0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No internet service	No internet service	Two year	No	Mailed check	25.75		1	0	No
0	Yes	Yes	0	No	No phone service	DSL	Yes	...	Yes	No	Two year	No	Credit card (automatic)	56.05		0	0	No
0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service	Two year	No	Mailed check	19.85		0	0	No
0	Yes	Yes	0	Yes	Yes	No	No internet service	...	No internet service	No internet service	Two year	No	Mailed check	25.35		0	0	No
0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service	Two year	No	Mailed check	20.00		5	0	No
0	Yes	Yes	0	Yes	No	No	No internet service	...	No internet service	No internet service	One year	Yes	Mailed check	19.70		0	0	No
0	Yes	Yes	0	Yes	Yes	DSL	No	...	Yes	No	Two year	No	Mailed check	73.35		0	0	No
0	No	Yes	0	Yes	Yes	DSL	Yes	...	No	No	Two year	Yes	Bank transfer (automatic)	61.90		0	0	No

Figura 22: Valores nulos TotalCharges. Autor: Blanca Mejía, 2024.

En la **Figura 23** se realiza programación para reemplazar por la mediana los valores nulos de la columna *"TotalCharges".*

```
# Reemplazar datos faltantes en la columna 'TotalCharges' con la mediana
Clientes['TotalCharges'] = Clientes['TotalCharges'].replace(' ', np.nan)
Clientes['TotalCharges'] = pd.to_numeric(Clientes['TotalCharges'])
Clientes['TotalCharges'].fillna(Clientes['TotalCharges'].median(), inplace=True)
```

Figura 23: Reemplazar nulos con mediana. Autor: Blanca Mejía, 2024.

En la **Figura 24** se revisa y verifica los valores después de reemplazar *TotalCharges* con mediana y además que todo el *dataset* se encuentre sin valores nulos.

```
# Verificar los valores despues de reemplazar TotalCharges con mediana
valores_nulos = Clientes.isnull().sum()
print("\nValores nulos:\n", valores_nulos)


Valores nulos:
 customerID          0
gender               0
SeniorCitizen        0
Partner              0
Dependents           0
tenure               0
PhoneService         0
MultipleLines        0
InternetService      0
OnlineSecurity       0
OnlineBackup         0
DeviceProtection     0
TechSupport          0
StreamingTV          0
StreamingMovies      0
Contract             0
PaperlessBilling     0
PaymentMethod        0
MonthlyCharges       0
TotalCharges         0
numAdminTickets      0
numTechTickets       0
Churn                0
dtype: int64
```

Figura 24: Revisión valores nulos. Autor: Blanca Mejía, 2024.

3.3.1.3. Redundancia de en los datos

Se observó una duplicidad de información en algunas variables, las mismas que se van a ajustar para reducir la complejidad y mejorar la claridad del conjunto de datos. Este ajuste tiene el propósito de disminuir dimensiones y simplificar la información disponible. En la **Figura 25** se observa el remplazo de todos los campos que tienen valores como *'No phone service':'No','No internet service':* se unifican a 'No'.

```
#REEMPLAZO VALORES REDUNDANTES QUE TIENEN EL CAMPO NO
Clientes.replace({'No phone service':'No','No internet service':'No'},inplace=True)
```

Figura 25: Variables redundantes. Autor: Blanca Mejía, 2024.

3.3.1.4. Redundancia de en los datos Construcción de la data

El *dataset* en su mayoría de parámetros se compone en su gran mayoría de variables categóricas, únicamente posee 3 variables de tipo numéricas, para lo cual se trabajará con las mismas variables sin generar nuevas variables. En la **Figura 26** se elimina la variable *customerID* porque no aporta información relevante para el modelo.

```
#Eliminar la columna customerID ya que no es reelevante para este estudio
Clientes = Clientes.drop('customerID', axis=1)
```

Figura 26: Eliminación customerID. Autor: Blanca Mejía, 2024.

Se ha realizado la transformación de las variables *SeniorCitizen, Partner, Dependents, PhoneService, PaperlessBilling, Churn* mediante los métodos de *hot encoding* y *dummy encoding* porque tienen variables de Yes y No.

El objetivo es preparar y mejorar la calidad de las variables categóricas mediante la codificación de etiquetas en un formato numérico binarios 1 y 0 mediante etiquetas (*Label Encoder*) como se observa en la **Figura 27**, dejando cada vez más pulido el *dataset* para la fase de modelado.

```
#Muestro las columnas
Clientes.columns

Index(['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure',
       'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod',
       'MonthlyCharges', 'TotalCharges', 'numAdminTickets', 'numTechTickets',
       'Churn'],
      dtype='object')

#Cambio para columnas tienen exclusivamente valores 'Yes' o 'No',
Yes_No_Columns = ['SeniorCitizen', 'Partner', 'Dependents', 'PhoneService', 'PaperlessBilling', 'Churn']

#Convierto variables categóricas en valores numéricas, asignando un número entero a cada categoría única.
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
for column in Yes_No_Columns:
    Clientes[column] = label_encoder.fit_transform(Clientes[column])
```

Figura 27: Codificación LabelEncoder. Autor: Blanca Mejía, 2024.

En este proceso se va a cambiar las variables dicotómicas de Yes y No y reemplazarlas por 1 y 0 (1=Si, 0=No).

En la **Figura 28** se observa que para el caso de las variables que son de tipo categóricas con más de dos clases se aplicó el método *dummies.*

```
#Convierto en Dummieslas columnas
for column in columns_to_encode:
    column_dummies = pd.get_dummies(Clientes[column], prefix=f'{column}_', dummy_na=False)
    Clientes = pd.concat([Clientes, column_dummies], axis=1)
```

Figura 28: Codificación de dummies. Autor: Blanca Mejía, 2024.

En la **Figura 29** se puede apreciar el resultado de las variables finales y con las cuales se va a trabajar en el modelo predictivo.

```
Clientes.columns
```

```
Index(['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure',
       'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',
       'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',
       'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod',
       'MonthlyCharges', 'TotalCharges', 'numAdminTickets', 'numTechTickets',
       'Churn', 'gender__Female', 'gender__Male', 'MultipleLines__No',
       'MultipleLines__Yes', 'InternetService__DSL',
       'InternetService__Fiber optic', 'InternetService__No',
       'OnlineSecurity__No', 'OnlineSecurity__Yes', 'OnlineBackup__No',
       'OnlineBackup__Yes', 'DeviceProtection__No', 'DeviceProtection__Yes',
       'TechSupport__No', 'TechSupport__Yes', 'StreamingTV__No',
       'StreamingTV__Yes', 'StreamingMovies__No', 'StreamingMovies__Yes',
       'Contract__Month-to-month', 'Contract__One year', 'Contract__Two year',
       'PaymentMethod__Bank transfer (automatic)',
       'PaymentMethod__Credit card (automatic)',
       'PaymentMethod__Electronic check', 'PaymentMethod__Mailed check'],
      dtype='object')
```

Figura 29: Variables finales del Dataset. Autor: Blanca Mejía, 2024.

3.3.1.5. Formato de la data

En la **Figura 30** se observa el proceso de estandarizar los datos con valores a Z-Scores, la variable *Churn* no se tomará en cuenta, ya que al ser la variable objetivo es importante mantenerla como dicotómica. El *dataset* ahora contiene todas las características estandarizadas. Este *dataset* puede ser utilizado para entrenar modelos de aprendizaje automático en el que la estandarización de características es importante para lograr buena precisión en los algoritmos de aprendizaje automático.

```
import pandas as pd
from sklearn.preprocessing import StandardScaler
columns_to_scale = [col for col in Clientes.columns if col != 'Churn']
Clientes_selected = Clientes[columns_to_scale]
churn_column = Clientes['Churn']
scaler = StandardScaler()
Clientes_st_sc = scaler.fit_transform(Clientes_selected)
Clientes_st_sc_df = pd.DataFrame(Clientes_st_sc, columns=columns_to_scale)
Clientes_st_sc_df['Churn'] = churn_column
print(Clientes_st_sc_df)
```

Figura 30: Características Estandarizadas. Autor: Blanca Mejía, 2024.

En la **Figura 31** se puede observar el resultado final después de realizar el proceso de estandarización del formato.

	SeniorCitizen	Partner	Dependents	tenure	PhoneService	\
0	-0.439916	1.034530	-0.654012	-1.277445	-3.054010	
1	-0.439916	-0.966622	-0.654012	0.066327	0.327438	
2	-0.439916	-0.966622	-0.654012	-1.236724	0.327438	
3	-0.439916	-0.966622	-0.654012	0.514251	-3.054010	
4	-0.439916	-0.966622	-0.654012	-1.236724	0.327438	
...	...	...	...	...	...	
7038	-0.439916	1.034530	1.529024	-0.340876	0.327438	
7039	-0.439916	1.034530	1.529024	1.613701	0.327438	
7040	-0.439916	1.034530	1.529024	-0.870241	-3.054010	
7041	2.273159	1.034530	-0.654012	-1.155283	0.327438	
7042	-0.439916	-0.966622	-0.654012	1.369379	0.327438	

	PaperlessBilling	MonthlyCharges	TotalCharges	numAdminTickets
0	0.829798	-1.160323	-0.994242	-0.404396
1	-1.205113	-0.259629	-0.173244	-0.404396
2	0.829798	-0.362660	-0.959674	-0.404396
3	-1.205113	-0.746535	-0.194766	-0.404396
4	0.829798	0.197365	-0.940470	-0.404396
...	...	...	...	...
7038	0.829798	0.665992	-0.128655	-0.404396
7039	0.829798	1.277533	2.243151	-0.404396
7040	0.829798	-1.168632	-0.854469	-0.404396
7041	0.829798	0.320338	-0.872062	-0.404396
7042	0.829798	1.358961	2.014288	1.163975

Figura 31: Proceso de estandarización. Autor: Blanca Mejía, 2024.

3.4. Modelado

En la **Figura 32** se observa la fase de Modelado. Una vez finalizada la fase de entendimiento y preparación de la data, se debe escoger los modelos sobre los cuales se aplicará el conjunto de entrenamiento para posteriormente evaluar el pronóstico y la precisión de estos. En este proyecto se realizarán varios tipos de modelos predictivos supervisados, varias técnicas de modelado son seleccionadas, aplicadas y sus parámetros son calibrados a valores óptimos.

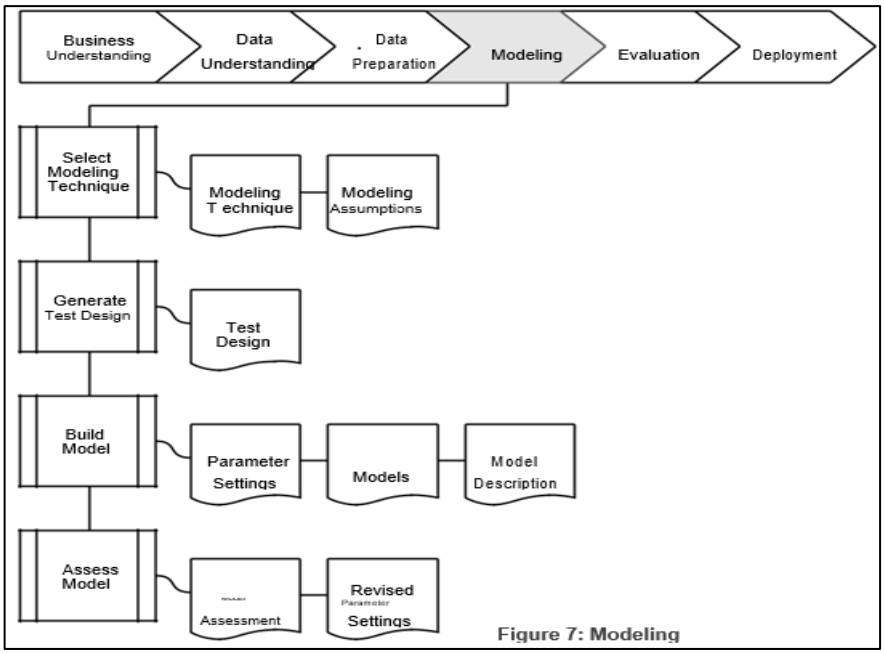


Figura 32: Fase de Modelado. Autor: Desconocido, recuperado 2024.

3.4.1. Selección de los modelos

El objetivo principal de este proyecto es realizar clasificación del *dataset* para predecir si los clientes cancelarán o no el servicio mediante el desarrollo de modelos de clasificación supervisados. Se consideró la aplicación de algoritmos de clasificación, como Regresión Logística, Árbol de decisión, Redes Neuronales, *Random Forest Classifier* y *Lazy Classifier*.

3.4.1.1. Modelo de Regresión Logística

La regresión logística ofrece simplicidad e interpretabilidad, lo que la convierte en una opción atractiva para muchas aplicaciones. Funciona bien en tareas de clasificación binaria, proporcionando predicciones probabilísticas entre 0 y 1, lo que la hace especialmente adecuada para escenarios como la predicción de compras de clientes, detección de fraudes, diagnósticos médicos, entre otras aplicaciones.

La regresión logística es el modelo que predice la probabilidad de que una muestra pertenezca a una clase determinada, la variable dependiente es binaria. La regresión es una prueba predictiva que se utiliza para describir la relación entre un conjunto de variables independientes y una variable binaria dependiente. Para el abandono de clientes, se ha utilizado una regresión logística para estimar la probabilidad de abandono en función del conjunto de caracteres o variables de los clientes. (*Sahu,* recuperado 2024*)*.

3.4.1.2. Modelo Árbol de decisión

“Un árbol de decisión es una estructura de árbol similar a un diagrama de flujo, donde cada nodo denota una prueba en un valor de atributo, cada rama representa un resultado de la prueba y las hojas del árbol representan clases o distribuciones de clases. Los árboles de decisión se pueden convertir fácilmente en reglas de clasificación”. (Jiawei Han,2006).

3.4.1.3. Modelo de Redes Neuronales

Las Redes de Neuronas Artificiales (denominadas habitualmente en castellano como RNA o en inglés como ANN, por Artificial Neural Networks) pueden definirse en primera aproximación como redes interconectadas masivamente en paralelo de elementos básicos muy simples de tipo software o hardware con organización jerárquica, capaces de adoptar un comportamiento colectivo adaptativo con el que intentan interactuar con los objetos del mundo real de modo análogo como lo hace el sistema nervioso biológico. (Blum, 1992).

3.4.1.4. Modelo de Random Forest Classifier

El término *random forest* se toma de la primera propuesta en el año 1995, este algoritmo está considerado como un clasificador bastante preciso. Trabaja bien, aunque haya datos perdidos y ofrece un método para la interacción de las variables (Breiman, 2001).

Es una técnica mejorada de *bagging*, que ayuda a obtener una precisión más alta en la clasificación mediante la incorporación de aleatoriedad en la construcción de cada clasificador individual. Esta aleatorización puede introducirse en la partición del espacio (construcción del árbol), así como en la muestra de entrenamiento. (Breiman, 2001).

3.4.1.5. Modelo de Lazy Classifier.

Lazy Classifier es tipo de modelo que no realiza un aprendizaje activo o significativo durante la fase de entrenamiento, simplemente memoriza los datos de entrenamiento y retiene esta información para realizar predicciones.

3.4.2. Matriz de Confusión

Este código es útil para evaluar el rendimiento del modelo de redes neuronales en el conjunto de datos de entrenamiento, particularmente en términos de exactitud y matriz de confusión. La matriz de confusión proporciona información detallada sobre cómo el modelo clasifica las instancias en cada clase.

3.4.3. Conceptos de evaluación de modelos predictivos

- **Exactitud (*Accuracy*):** La exactitud es una medida de la corrección general del modelo. Se calcula como la proporción de instancias correctamente predichas con respecto al total de instancias.

- **Precisión (*Precision*):** La precisión es la proporción de observaciones positivas correctamente predichas con respecto al total de positivos predichos. Es una medida de cuántas de las instancias positivas predichas son realmente positivas.

- **Recuperación (*Recall* o Sensibilidad):** La recuperación es la proporción de observaciones positivas correctamente predichas con respecto a todas las observaciones en la clase real.

- **Especificidad (Tasa de Verdaderos Negativos):** La especificidad es la proporción de observaciones negativas correctamente predichas con respecto al total de

observaciones en la clase real negativa.

- **Puntuación F1:** La puntuación F1 es la media armónica de la precisión y la recuperación. Proporciona un equilibrio entre precisión y recuperación, especialmente útil cuando hay una distribución desigual entre las clases.

a. Generar diseño de prueba

En esta etapa lo que se quiere es considerar lo que tiene que ver con los temas relacionados a la prueba o *testing* del modelo, se desea conseguir métricas asociadas a la matriz de confusión que indiquen buenas capacidades predictivas, En la **Figura 33** se observa la cantidad de registros con Yes (1) y No (0) que existe en la variable objetivo "*Churn*", dado que lo que se requiere es poder identificar los condicionantes que hacen que los clientes cancelen el servicio.

```
Clientes['Churn'].value_counts()

0    5174
1    1869
Name: Churn, dtype: int64
```

Figura 33: Variable objetivo "Churn". Autor: Blanca Mejía, 2024.

b. Construcción del modelo

Como se observa en la **Figura 34** se realiza la declaración de las respectivas librerías necesarias para todos los procesos y algoritmos implementados en los diferentes modelos predictivos.

```
[2]  # Librerías
     import numpy as np
     import pandas as pd
     from pandas import read_csv
     from sklearn.model_selection import train_test_split
     from sklearn.metrics import classification_report
     from sklearn.metrics import confusion_matrix
     from sklearn.metrics import accuracy_score
     import seaborn as sns
     import matplotlib.pyplot as plt
     from sklearn.model_selection import train_test_split
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
     from sklearn import metrics
     from sklearn.metrics import mean_squared_error, r2_score
     from sklearn.datasets import load_wine
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import RocCurveDisplay
     from sklearn.model_selection import train_test_split
     from sklearn.svm import SVC
     import missingno as msno
     import plotly.express as px
     import plotly.graph_objects as go
     from plotly.subplots import make_subplots
     from sklearn.metrics import precision_score
     from sklearn.metrics import recall_score
     from sklearn.metrics import f1_score
     from sklearn.metrics import confusion_matrix
     from sklearn.metrics import roc_curve, auc
     from sklearn.preprocessing import StandardScaler
     from sklearn import linear_model

     %matplotlib inline
```

Figura 34: Declaración de librerías. Autor: Blanca Mejía, 2024.

c. Parametrización de variables de entrenamiento

En la **Figura 35** se realiza la separación de la variable objetivo de las que serían las variables predictoras. Este conjunto de entrenamiento serán las variables predictoras y la variable objetivo con las cuales se entrenarán todos los modelos.

DECLARACIÓN DE CONJUNTO DE PARAMETROS DE TRAIN Y TEST

```
from sklearn.model_selection import train_test_split
#BORRO LA COLUMNA CHURN Y REEMPLAZO POR 1 Y 0
X = Clientes_st_sc_df.drop('Churn', axis=1)
y = Clientes_st_sc_df['Churn']
```

Figura 35: Variables de Train y Test. Autor: Blanca Mejía, 2024.

Se trabajará con los hiperparámetros por defecto cada modelo, considerando, por ejemplo, un número máximo de iteraciones de 100 y la selección de un problema binario. Estas funciones *train* y *test* permiten dividir al conjunto de datos en dos bloques de entrenamiento y prueba del modelo, por ello se denominan *train* (entrenamiento) y *test* (pruebas).

El conjunto de entrenamiento del modelo es necesario para entrenamiento, mientras que el conjunto de pruebas se usa para evaluar el desempeño en datos. Como resultado se obtiene que tan funcional es el modelo y que tanto ha aprendido los patrones y muestra una muestra preliminar de como funcionara con una data de mayor volumen.

En la **Figura 36** se observa como el código divide su conjunto de datos (X, y) en un conjunto de entrenamiento (70%) y un conjunto de prueba (30%) por eso el *test_size* es igual a 0.3. El hiperparametro *randon_state* busca conseguir cierta repetición con los resultados. Se utiliza para inicializar el generador interno de números aleatorios, que decidirá la división de los datos en índices de *train* y *test*. Esto es para garantizar la reproducibilidad.

```
#CONJUNTO DE ENTRENAMIENTO Y PRUEBA
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,
                                                    random_state=1, stratify=y)
```

Figura 36: Conjunto de entrenamiento y prueba. Autor: Blanca Mejía, 2024.

El parámetro *stratify* de la función *train_test_split* se utiliza para garantizar que la división de los datos mantenga la misma distribución de la variable objetivo (y) en los conjuntos de entrenamiento y de prueba que en el conjunto de datos original.

Esto es especialmente importante cuando se trata de conjuntos de datos desequilibrados como sucede en este caso, en los que una clase puede estar significativamente infrarrepresentada.

3.5. Evaluación

En la **Figura 37** se observa la fase de evaluación, esta fase el modelo ya se pone en marcha de forma iterativa y repetitiva, se obtienen los resultados de los modelos construidos, en esta fase es muy importante evaluar uno a uno los resultados y pasos ejecutados para empezar a comparar si los resultados que se obtuvieron están alineados con los objeticos del proyecto y con los objetivos del negocio.

Para este proyecto, se creó un modelo que permite predecir la variable objetivo '*Churn'* de acuerdo con características particulares de las observaciones, en la siguiente etapa se procederá a evaluar a partir de la matriz de confusión generada en cada uno de los modelos correspondientes.

Se realizó la construcción de los modelos Redes Neuronales, *Classification Report*, Árbol de decisión, *Random Forest Classifier* y *Lazy Classifier.*

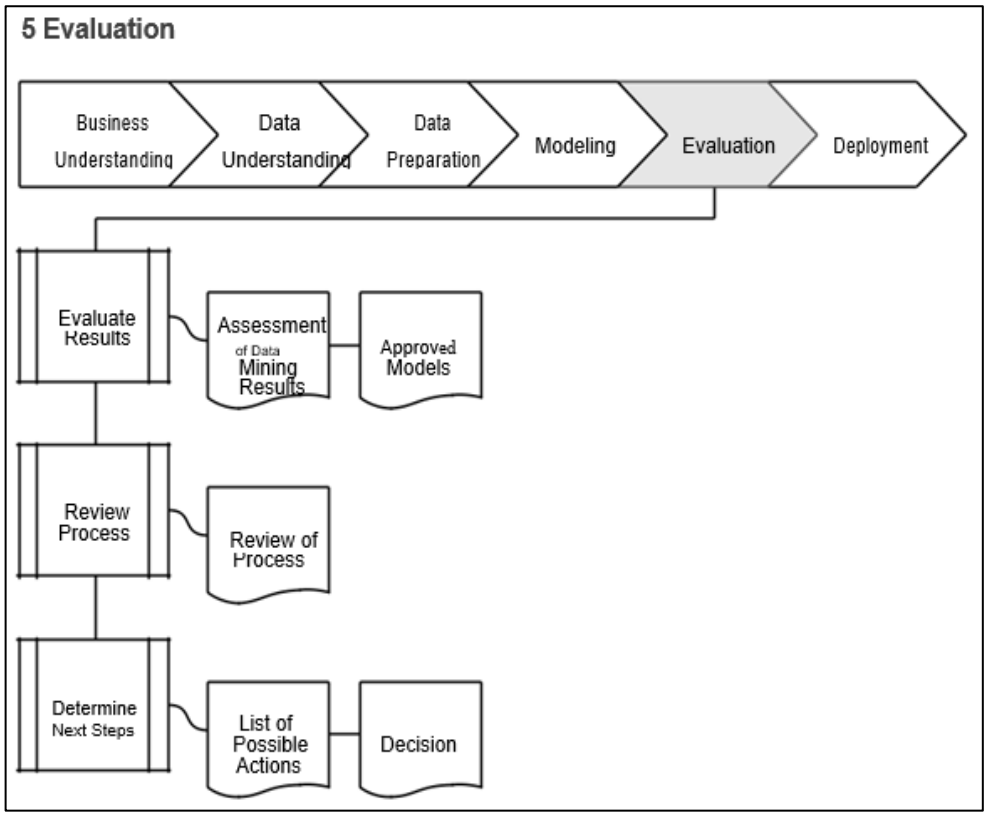


Figura 37: Fase de Evaluación. Autor: Desconocido, recuperado 2024.

3.5.1. Modelo Regresión Logística

Como se muestra en la **Figura 38** se procede a construir y ejecutar el modelo de regresión logística con las variables de entrenamiento y *test* para obtener las predicciones sobre la variable objetivo '*Churn'.* En la opción *print*, imprime las matrices obtenidas en la variable de prueba y la variable predictora.

1) Algoritmo Regresión Logística

```
from sklearn import linear_model
Classifier = linear_model.LogisticRegression()
Classifier.fit(X_train, y_train)
# make predictions for test data
y_pred = Classifier.predict(X_test)

print(list(y_test))
print(list(y_pred))
```

```
[0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1,
[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,
```

Figura 38: Modelo de Regresión Logística. Autor: Blanca Mejía, 2024.

Los conjuntos de datos de los clientes son analizados y realizan ecuaciones de regresión. Posteriormente se realiza una fase de evaluación para cada cliente que conforma el conjunto de datos.

3.5.1.1. Matriz de confusión

En la **Figura 39** se puede observar la matriz de confusión que indica que los valores de la diagonal principal 1423 y 403 corresponden con los valores estimados de forma correcta por el modelo ya que los valores más altos deben ser los de la diagonal principal, en este caso si cumple. Las variables de tipo dicotómico (Yes/No) lo que significa que los valores asignados serán 0 y 1 (0 para No y 1 para Si).

- Si cancela (1).
- No cancela (0).

Verdaderos Positivos (TP): 1423.
Verdaderos Negativos (TN): 403.
Falsos Positivos (FN): 129.
Falsos Negativos (FP): 158.

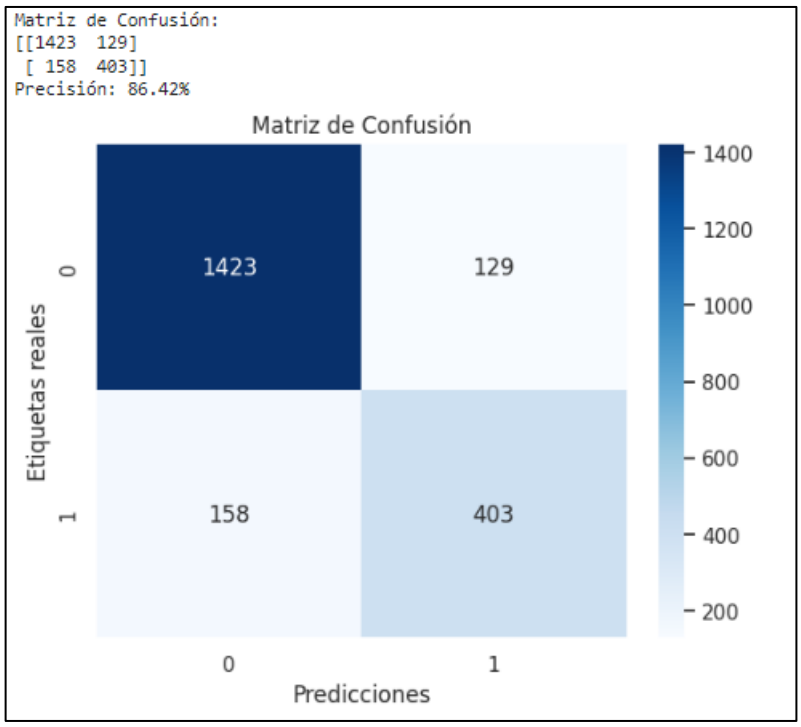


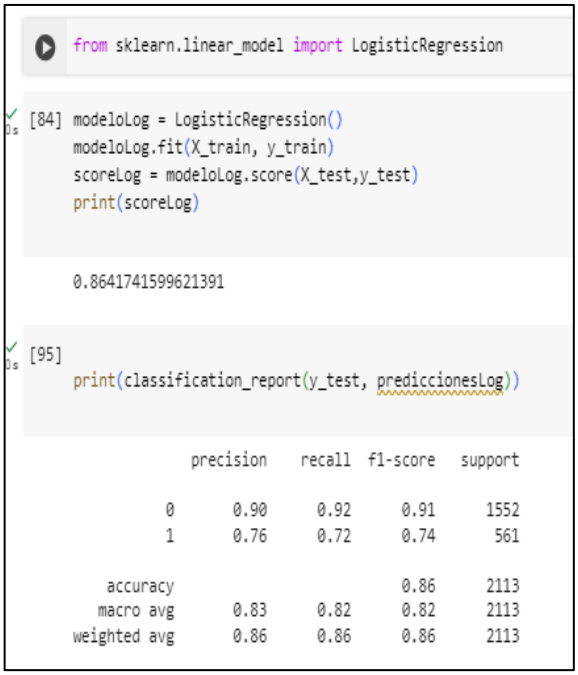
Figura 39: Matriz de confusión Regresión Logística. Autor: Blanca Mejía, 2024.

En las **Figuras 40 y 41** se puede observar los resultados obtenidos del modelo de Regresión logística que se detallaran uno a uno más adelante.

```
from sklearn.metrics import confusion_matrix

# Calculate confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Extract values from confusion matrix
tn, fp, fn, tp = conf_matrix.ravel()

# Calculate accuracy
accuracy = (tp + tn) / (tp + tn + fp + fn)

# Calculate precision
precision = tp / (tp + fp)

# Calculate recall
recall = tp / (tp + fn)

specificity = tn / (tn + fp)

# Calculate F1 score
f1 = 2 * (precision * recall) / (precision + recall)

# Print the metrics
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("Specificity:", specificity)
print("F1 Score:", f1)
```

```
Accuracy: 0.8641741599621391
Precision: 0.7575187969924813
Recall: 0.7183600713012478
Specificity: 0.916881443298969
F1 Score: 0.7374199451052149
```

Figura 40: Resultados Regresión Logística. Autor: Blanca Mejía, 2024.



```
from sklearn.linear_model import LogisticRegression

[84] modeloLog = LogisticRegression()
     modeloLog.fit(X_train, y_train)
     scoreLog = modeloLog.score(X_test,y_test)
     print(scoreLog)

     0.8641741599621391

[95]
     print(classification_report(y_test, prediccionesLog))

                  precision    recall  f1-score   support

               0       0.90      0.92      0.91      1552
               1       0.76      0.72      0.74       561

        accuracy                           0.86      2113
       macro avg       0.83      0.82      0.82      2113
    weighted avg       0.86      0.86      0.86      2113
```

Figura 41: Resultados Regresión Logística. Autor: Blanca Mejía, 2024.

3.5.1.2. Resultados obtenidos en el modelo de Regresión Logistica:

- **Precisión:**

  Si Cancela (1) el porcentaje de acierto es 0.76%.
  No Cancela (0) el porcentaje de acierto es 0.90%.
  El modelo tiene una precisión es del 76%. Esto indica que, de todas las instancias predichas como positivas, el 76% realmente son positivas. Una precisión relativamente alta sugiere que cuando el modelo predice una clase, es probable que sea correcta.

- ***Recall:***

  Si Cancela (1) el porcentaje de acierto es 0.72%.
  No Cancela (0) el porcentaje de acierto es 92%.
  El modelo tiene una recuperación del 72%. Esto significa que el modelo captura el 72% de todas las instancias positivas reales. Una recuperación moderada indica que el modelo no identifica todas las instancias positivas, pero captura más de la mitad.

- **F1-score:**

  Si Cancela (1) el porcentaje de acierto es 0.74%.

No Cancela (0) el porcentaje de acierto es 91%.
El modelo tiene una puntuación F1 del 74%. Un valor relativamente alto sugiere que el modelo está encontrando un buen compromiso entre la precisión y la recuperación, en este modelo supera el 50%.

- **Accuracy:** 0.86%

Dada la naturaleza dicotómica de la variable objetivo Churn, el modelo de Regresión Logística indica un 86%.
En el modelo tiene una exactitud del 86%. Esto significa que aproximadamente el 86% de todas las predicciones (positivas y negativas) son correctas.

- **Especificidad (Tasa de Verdaderos Negativos):**

El modelo tiene una especificidad del 92%. Un valor alto de especificidad (cercano a 1) sugiere que el modelo es bueno para predecir instancias negativas.

- **Resumen:**

La exactitud indica que el modelo está acertando en aproximadamente el 86% de las instancias, lo cual es bastante razonable.

La precisión y la recuperación con aproximadamente 76% indica que son moderadas, que el modelo tiene un equilibrio aceptable entre identificar instancias positivas y evitar falsos positivos.

La especificidad es alta, aproximadamente 72% indicando que el modelo es eficaz para predecir instancias negativas.

La puntuación F1 sugiere un buen equilibrio, aproximadamente 74% entre precisión y recuperación.

En conclusión, el modelo tiene un rendimiento generalmente aceptable, el cual deberá ser acoplado al alcance del problema que se desea solucionar con el modelo predictivo para conocer los clientes que cancelarán el servicio.

3.5.1.3. CURVA ROC:

En la **Figura 42** se observa la curva ROC y áreas bajo la curva, en este modelo indica que el modelo puede distinguir en un 82% de capacidad predictiva.

```
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

fpr, tpr, lim = roc_curve(y_test, y_pred)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='red', lw=2, label=f'Curva ROC (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("Tasa de Falsos Positivos")
plt.ylabel("Tasa de Verdaderos Positivos")
plt.title("Curva ROC")
plt.legend(loc="lower right")
plt.show()
```

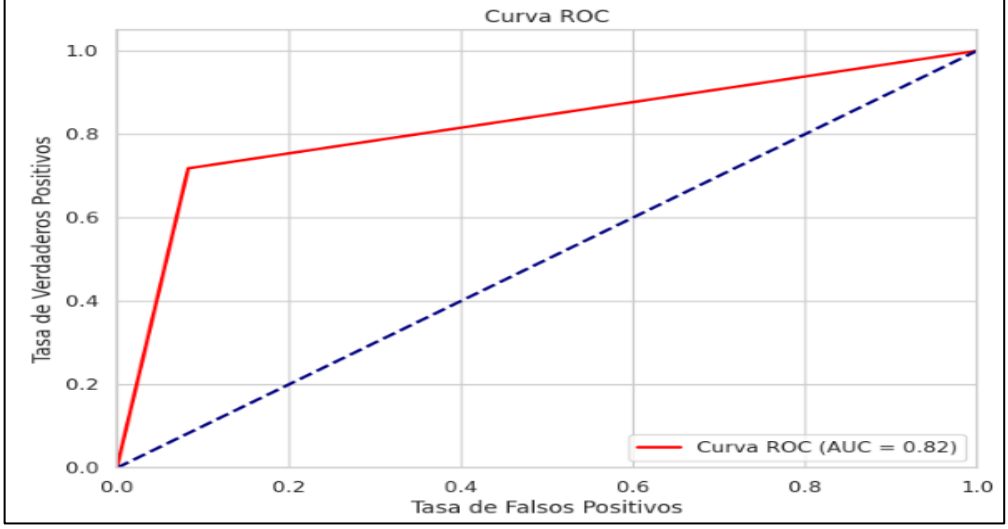


Figura 42: Curva ROC Regresión Logística. Autor: Blanca Mejía, 2024.

3.5.2. Modelo Árbol de decisión

- En la **Figura 43** inicialmente se declara la librería *from sklearn import tree* para crear clasificadores de árbol de decisión.
- Se crea un objeto clasificador de decisión llamado **árbol** con criterio de entropía para la división del árbol de decisión, *criterion='entropy'*.
- La profundidad del árbol es de 5 niveles, *max_depth=5*.
- El número mínimo de muestras requeridas para dividir un nodo interno *min_samples_split* igual a 2.
- El número mínimo de muestras requeridas para estar en un nodo hoja *min_samples_leaf* es 1.
- Luego se declaran los datos de entrenamiento de X_train y la etiqueta y_train.

```
[274] from sklearn import tree
      arbol = tree.DecisionTreeClassifier(
          criterion = 'entropy',
          max_depth= 5,
          min_samples_split=2,
          min_samples_leaf = 1
      )
      arbol = arbol.fit(X_train, y_train)

[275] tree.plot_tree(arbol, filled = True)
      plt.show()
```

Figura 43: Construcción Árbol de decisión. Autor: Blanca Mejía, 2024.

En la **Figura 44** se observa la construcción del modelo de árbol de decisión, dada la magnitud del gráfico no es factible observar claramente los nodos e información, lo importante es que si se distingue y comprende el proceso de clasificación.

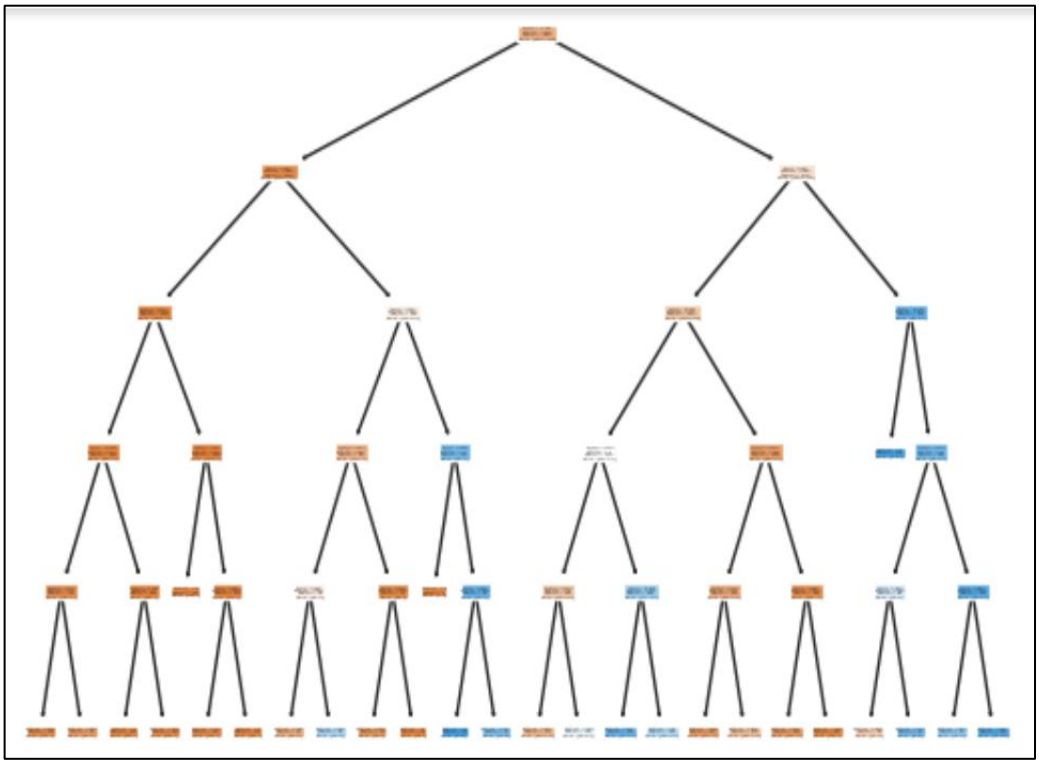


Figura 44: Construcción árbol de decisión. Autor: Blanca Mejía, 2024.

3.5.2.1. Matriz de confusión

En la **Figura 45** se construye la matriz de confusión para evaluar los resultados del modelo Árbol de decisión.

```
# Predicciones
predicciones = arbol.predict(X_train)
print(predicciones)
# Comparativa
exactitud = accuracy_score(y_train,predicciones)
print(exactitud)
# Matriz de confusión
matriz_conf = confusion_matrix(y_train, predicciones)
print(matriz_conf)

sns.set()
#Generación de clases
classes = ['Si', 'No']
sns.heatmap(matriz_conf, annot=True, fmt = 'd', cmap = 'Blues', xticklabels=classes,
            yticklabels= classes)
plt.title("Matriz de confusión")
plt.xlabel("Etiqueta predicha")
plt.ylabel("Etiqueta Real")
plt.show()
```

Figura 45: Matriz de confusión Arboles de decisión. Autor: Blanca Mejía, 2024.

En la **Figura 46** se puede observar la matriz de confusión que indica que los valores de la diagonal principal 3289 y 916 corresponden con los valores estimados de forma correcta por el modelo ya que los valores más altos deben ser los de la diagonal principal, en este caso si cumple. Las variables de tipo dicotómico (Yes/No) lo que significa que los valores asignados serán 0 y 1 (0 para No y 1 para Si).

- Si cancela (1).
- No cancela (0).

Verdaderos Positivos (TP): 3289.
Verdaderos Negativos (TN): 916.
Falsos Positivos (FN): 333.
Falsos Negativos (FP): 392.

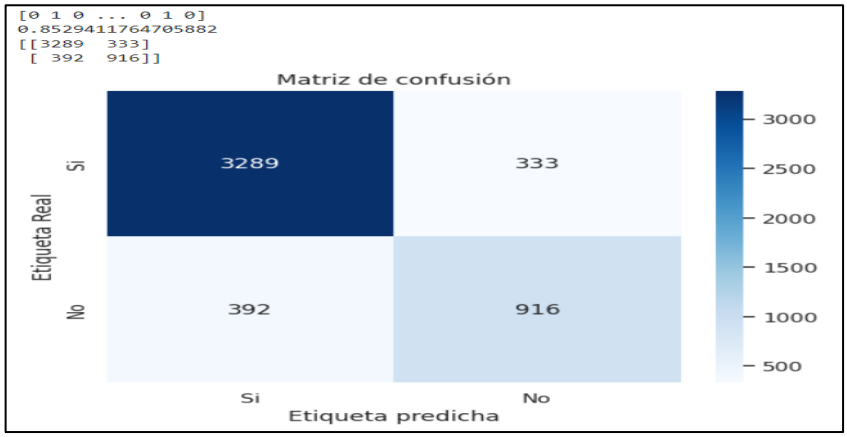


Figura 46: Matriz de confusión Árbol de decisión. Autor: Blanca Mejía, 2024.

En la **Figuras 47** se puede observar los resultados obtenidos del modelo Árbol de decisión que se detallaran uno a uno más adelante.

```
exactitud = accuracy_score(y_train,predicciones)
print('Exactitud: ',exactitud*100)
precision = precision_score(y_train,predicciones)
print('Precisión: ', precision*100)
sensibilidad = recall_score(y_train,predicciones)
print('Recall: ', sensibilidad*100)
puntaje = f1_score(y_train,predicciones)
print('Puntaje: ', puntaje*100)
matriz_Tree = confusion_matrix(y_train, predicciones)
matriz_Tree
```

```
Exactitud:  85.29411764705883
Precisión:  73.3386709367494
Recall:  70.03058103975535
Puntaje:  71.64646069612827
array([[3289,  333],
       [ 392,  916]])
```

```
print(classification_report(y_train,predicciones))
```

```
              precision    recall  f1-score   support

           0       0.89      0.91      0.90      3622
           1       0.73      0.70      0.72      1308

    accuracy                           0.85      4930
   macro avg       0.81      0.80      0.81      4930
weighted avg       0.85      0.85      0.85      4930
```

Figura 47: Resultados Modelo Árbol de decisión. Autor: Blanca Mejía, 2024.

3.5.2.2. Resultados obtenidos en el modelo de Árbol de decisión:

- **Precisión:**

  Si Cancela (1) el porcentaje de acierto es 0.73%.

  No Cancela (0) el porcentaje de acierto es 0.89%.

  El modelo tiene una precisión es del 73%. Esto indica que, de todas las instancias predichas como positivas, el 73% realmente son positivas. Una precisión relativamente alta sugiere que cuando el modelo predice una clase, es probable que sea correcta.

- ***Recall:***

  Si Cancela (1) el porcentaje de acierto es 0.70%.

  No Cancela (0) el porcentaje de acierto es 0.91%.

  El modelo tiene una recuperación del 70%. Esto significa que el modelo captura el 70% de todas las instancias positivas reales. Una recuperación moderada indica que el modelo no identifica todas las instancias positivas, pero captura más de la mitad.

- ***F1-score:***

  Si Cancela (1) el porcentaje de acierto es 0.72%.
  No Cancela (0) el porcentaje de acierto es 0.90%.
  El modelo tiene una puntuación F1 del 72%. Un valor relativamente alto sugiere que el modelo está encontrando un buen compromiso entre la precisión y la recuperación, en este modelo supera el 50%.

- ***Accuracy:*** *0.85%*

  Dada la naturaleza dicotómica de la variable objetivo Churn, el modelo de árbol de decisión indica un 85%.
  En el modelo tiene una exactitud del 85%. Esto significa que aproximadamente el 85% de todas las predicciones (positivas y negativas) son correctas.

- **Resumen:**
  La exactitud indica que el modelo está acertando en aproximadamente el 85% de las instancias, lo cual es bastante razonable.

  La precisión y la recuperación con aproximadamente 73% indica que son moderadas, que el modelo tiene un equilibrio aceptable entre identificar instancias positivas y evitar falsos positivos.

  La puntuación F1 sugiere un buen equilibrio, aproximadamente 72% entre precisión y recuperación.

  En conclusión, el modelo tiene un rendimiento generalmente aceptable, el cual deberá ser acoplado al alcance del problema que se desea solucionar con el modelo predictivo para conocer los clientes que cancelarán el servicio.

3.5.2.3. CURVA ROC:

En la **Figura 48** se observa la curva ROC y áreas bajo la curva, en este modelo indica que el modelo puede distinguir en un 80% de capacidad predictiva.

```python
#CURVA ROC
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
fpr, tpr, lim = roc_curve(y_train, predicciones)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='red', lw=2, label=f'Curva ROC (AUC = {roc_auc:.2f})') # Ajuste en la etiqueta
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("Tasa de Falsos Positivos")
plt.ylabel("Tasa de Verdaderos Positivos")
plt.title("Curva ROC en Datos de Entrenamiento")
plt.legend(loc="lower right")
plt.show()
```

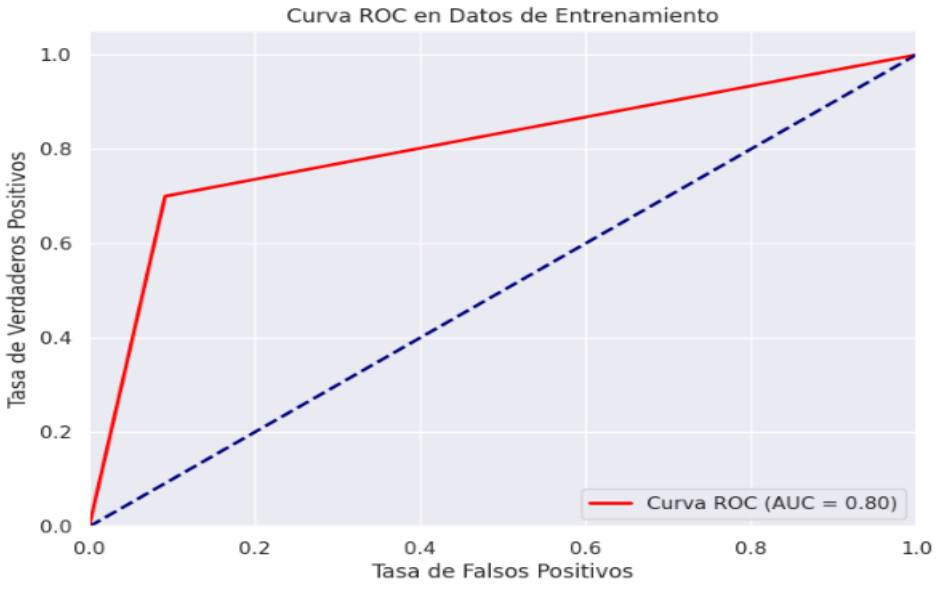


Figura 48: Curva ROC Modelo árbol de decisión. Autor: Blanca Mejía, 2024.

3.5.3. Modelo Redes Neuronales

- En la **Figura 49** inicialmente se declara la scikit-learn para crear un clasificador de redes neuronales artificiales utilizando la clase MLPClassifier para crear clasificadores clasificador de redes neuronales artificiales.
- Se establecen las iteraciones random_state igual a 1 fija la variable para generar números aleatorios, lo que permite que los resultados sean reproducibles.
- Se define un número máximo de iteraciones (épocas) durante el entrenamiento de la red neuronal max_iter. En este caso, se ha fijado en 300 iteraciones.
- Se declaran las características X_train representa las características de entrenamiento y y_train son las etiquetas de entrenamiento.

```
[316] from sklearn.neural_network import MLPClassifier
      red = MLPClassifier(random_state=1, max_iter=300).fit(X_train, y_train)

      /usr/local/lib/python3.10/dist-packages/sklearn/neural_network/_multilayer_perceptron.py:686: Conv
        warnings.warn(
```

Figura 49: Construcción Redes Neuronales. Autor: Blanca Mejía, 2024.

3.5.3.1. Matriz de confusión

En la **Figura 50** se construye la matriz de confusión para evaluar los resultados del modelo Redes Neuronales. En la **Figura 50** se puede observar la matriz de confusión que indica que los valores de la diagonal principal 1399 y 378 corresponden con los valores estimados de forma correcta por el modelo ya que los valores más altos deben ser los de la diagonal principal, en este caso si cumple. Las variables de tipo dicotómico (Yes/No) lo que significa que los valores asignados serán 0 y 1 (0 para No y 1 para Si).

- Si cancela (1).
- No cancela (0).

Verdaderos Positivos (TP): 1399.
Verdaderos Negativos (TN): 378.
Falsos Positivos (FN): 153.
Falsos Negativos (FP): 183.

```
classes = ['Si', 'No']
sns.heatmap(matriz_conf, annot=True, fmt = 'd', cmap = 'Blues', xticklabels=classes,
            yticklabels= classes)
plt.title("Matriz de confusión")
plt.xlabel("Etiqueta predicha")
plt.ylabel("Etiqueta Real")
plt.show()
```

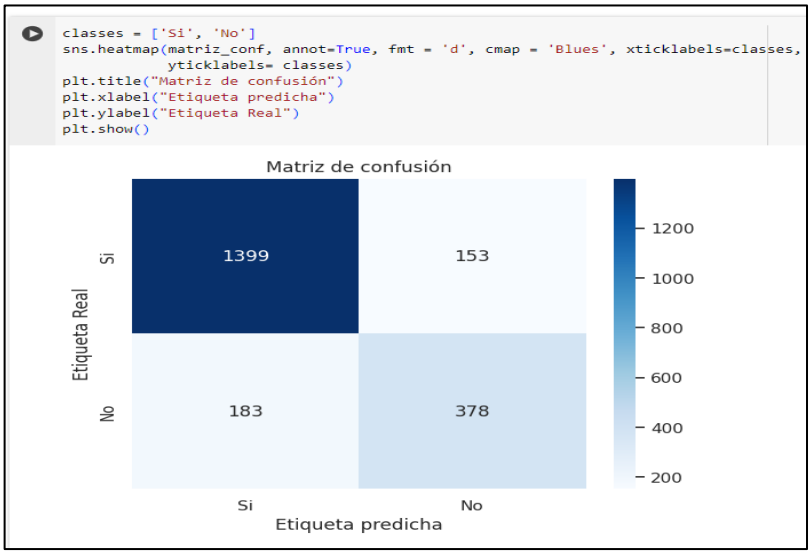


Figura 50: Matriz de confusión Redes Neuronales. Autor: Blanca Mejía, 2024.

En las **Figuras 51** se puede observar los resultados obtenidos del modelo Árbol de decisión que se detallaran uno a uno más adelante.

```
predicciones_nn = red.predict(X_test)
# Comparativa
exactitud = accuracy_score(y_test,predicciones_nn)
redes_exactitud=exactitud
print(redes_exactitud)
# Matriz de confusión
matriz_conf = confusion_matrix(y_test, predicciones_nn)
print(matriz_conf)
```

```
0.8409843823946995
[[1399  153]
 [ 183  378]]
```

```
from sklearn.metrics import classification_report
y_pred = red.predict(X_test)
# Imprimir el informe de clasificación
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.88      0.90      0.89      1552
           1       0.71      0.67      0.69       561

    accuracy                           0.84      2113
   macro avg       0.80      0.79      0.79      2113
weighted avg       0.84      0.84      0.84      2113
```

Figura 51: Resultados Redes Neuronales. Autor: Blanca Mejía, 2024.

3.5.3.2. Resultados obtenidos en el modelo de Redes Neuronales

- **Precisión:**

  Si Cancela (1) el porcentaje de acierto es 0.71%.

  No Cancela (0) el porcentaje de acierto es 0.88%.

  El modelo tiene una precisión es del 71%. Esto indica que, de todas las instancias predichas como positivas, el 71% realmente son positivas. Una precisión relativamente alta sugiere que cuando el modelo predice una clase, es probable que sea correcta.

- **Recall:**

  Si Cancela (1) el porcentaje de acierto es 0.67%.

  No Cancela (0) el porcentaje de acierto es 0.90%.

  El modelo tiene una recuperación del 67%. Esto significa que el modelo captura el 67% de todas las instancias positivas reales. Una recuperación moderada indica que el modelo no identifica todas las instancias positivas, pero captura más de la mitad.

- **F1-score:**

  Si Cancela (1) el porcentaje de acierto es 0.69%.
  No Cancela (0) el porcentaje de acierto es 0.89%.
  El modelo tiene una puntuación F1 del 69%. Un valor relativamente alto sugiere que el modelo está encontrando un buen compromiso entre la precisión y la recuperación, en este modelo supera el 50%.

- ***Accuracy:*** *0.84%*

  Dada la naturaleza dicotómica de la variable objetivo *Churn*, el modelo de árbol de decisión indica un 84%.
  En el modelo tiene una exactitud del 84%. Esto significa que aproximadamente el 84% de todas las predicciones (positivas y negativas) son correctas.

- **Resumen:**
  La exactitud indica que el modelo está acertando en aproximadamente el 84% de las instancias, lo cual es bastante razonable.

  La precisión y la recuperación con aproximadamente 71% indica que son moderadas, que el modelo tiene un equilibrio aceptable entre identificar instancias positivas y evitar falsos positivos.

  La puntuación F1 sugiere un buen equilibrio, aproximadamente 69% entre precisión y recuperación.

  En conclusión, el modelo tiene un rendimiento generalmente aceptable, el cual deberá ser acoplado al alcance del problema que se desea solucionar con el modelo predictivo para conocer los clientes que cancelarán el servicio.

3.5.3.3. CURVA ROC:

En la **Figura 52** se observa la curva ROC y áreas bajo la curva, en este modelo indica que el modelo puede distinguir en un 79% de capacidad predictiva.

```
#CURVA ROC
fpr, tpr, lim = roc_curve(y_test, predicciones_nn)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='red', lw=2, label=f'Curva ROC (AUC = {roc_auc:.2f})') # Ajuste en la etiqueta
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("Tasa de Falsos Positivos")
plt.ylabel("Tasa de Verdaderos Positivos")
plt.title("Curva ROC en Datos de Prueba")
plt.legend(loc="lower right")
plt.show()
```

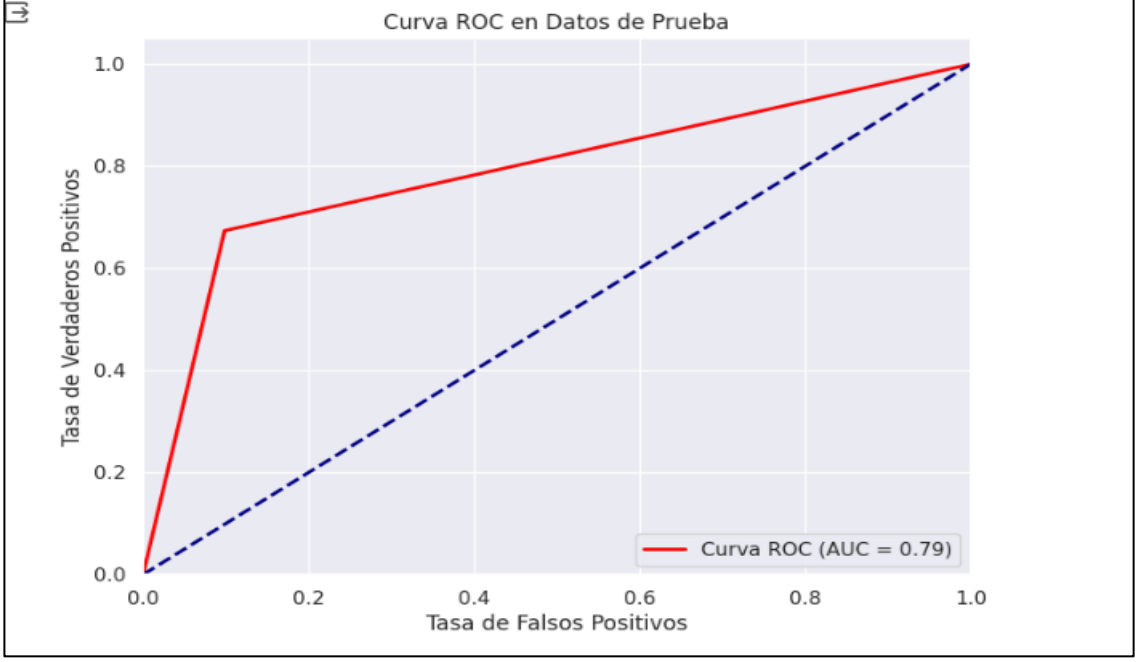


Figura 52: Curva ROC Redes Neuronales. Autor: Blanca Mejía, 2024.

3.5.4. Modelo Random Forest Classifier

- En la **Figura 53** inicialmente se declara la librería scikit-learn para crear y entrenar un clasificador RandomForest.
- Se declara el modelo RandomForestClassifier con un n_estimators igual a 100 número de árboles en el bosque.
- En el hiperparametro bootstrap se utiliza un muestreo con reemplazo para construir los árboles.
- En el hiperparametro verbose muestra información detallada durante el entrenamiento, en este caso es igual a 2.
- Finalmente, en max_features se define el número máximo de features que cada árbol puede utilizar.

```
[284] from sklearn.ensemble import RandomForestClassifier

      # Crear el modelo con 100 arboles
      model = RandomForestClassifier(n_estimators=100,
                                     bootstrap = True, verbose=2,
                                     max_features = 'sqrt')
      # a entrenar!
      model.fit(X_train, y_train)
      building tree 48 of 100
      building tree 49 of 100
      building tree 50 of 100
      building tree 51 of 100
      building tree 52 of 100
      building tree 53 of 100
      building tree 54 of 100
      building tree 55 of 100
      building tree 56 of 100
      building tree 57 of 100
      [Parallel(n_jobs=1)]: Done  40 tasks      | elapsed:    0.4s
      building tree 58 of 100
      building tree 59 of 100
      building tree 60 of 100
      building tree 61 of 100
      building tree 62 of 100
      building tree 63 of 100
      building tree 64 of 100
      building tree 65 of 100
      building tree 66 of 100
      building tree 67 of 100
      building tree 68 of 100
      building tree 69 of 100
      building tree 70 of 100
      building tree 71 of 100
      building tree 72 of 100
      building tree 73 of 100
      building tree 74 of 100
      building tree 75 of 100
      building tree 76 of 100
      building tree 77 of 100
      building tree 78 of 100
      building tree 79 of 100
      building tree 80 of 100
      building tree 81 of 100
      building tree 82 of 100
      building tree 83 of 100
```

Figura 53: Construcción Random Forest Classifier. Autor: Blanca Mejía, 2024.

En la **Figura 54** se observa en proceso de entrenamiento y las predicciones del modelo *Random Forest Classifier* que fue entrenado con 100 estimadores.

```
#RandomForestClassifier
forest = RandomForestClassifier()
forest.fit(X_train, y_train)

▾ RandomForestClassifier
RandomForestClassifier()

#REALIZAR PREDICCIONES
y_pred_test = forest.predict(X_test)

#VER EL ACCURACY
accuracy_score(y_test, y_pred_test)

0.8551822053951728

#IMPRIMIENDO LA PREDICCION
random=(metrics.accuracy_score(y_test, y_pred_test))
print (random)

0.8551822053951728

svc = SVC(random_state=42)
svc.fit(X_train, y_train)

▾ SVC
SVC(random_state=42)
```

Figura 54: Entrenamiento Random Forest Classifier. Autor: Blanca Mejía, 2024.

3.5.4.1. Matriz de confusión

En la **Figura 55** se construye la matriz de confusión para evaluar los resultados del modelo *Random Forest Classifier*.

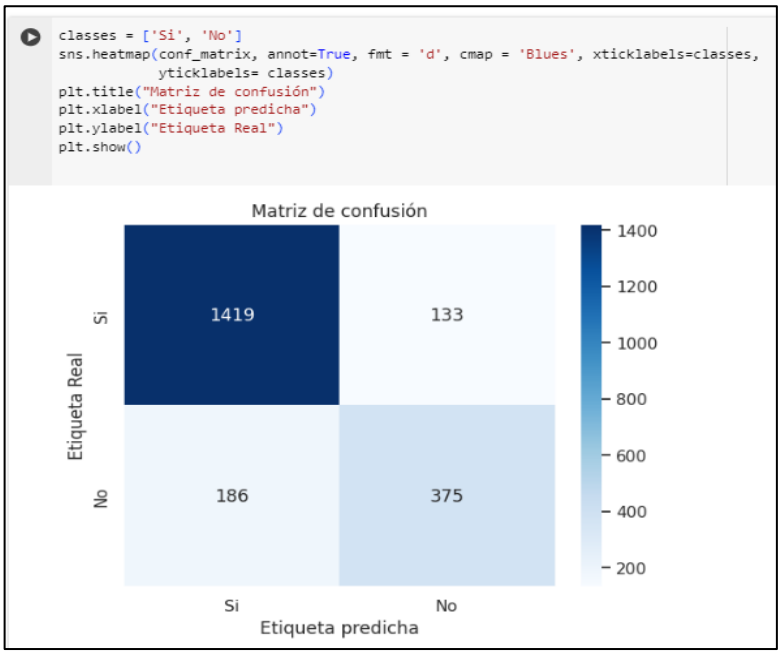


Figura 55: Matriz de confusión Random Forest Classifier. Autor: Blanca Mejía, 2024.

En la **Figura 55** se puede observar la matriz de confusión que indica que los valores de la diagonal principal 1419 y 375 corresponden con los valores estimados de forma correcta por el modelo ya que los valores más altos deben ser los de la diagonal principal, en este caso si cumple. Las variables de tipo dicotómico (Yes/No) lo que significa que los valores asignados serán 0 y 1 (0 para No y 1 para Si).

- Si cancela (1).
- No cancela (0).

Verdaderos Positivos (TP): 1419.
Verdaderos Negativos (TN): 375.
Falsos Positivos (FN): 133.
Falsos Negativos (FP): 186.

En la **Figura 56** se puede observar los resultados obtenidos del modelo *Random Forest Classifier* que se detallarán uno a uno más adelante.

```
[320] #MATRIZ DE CONFUSION DE LAS VARIABLES PREDICTORAS Y DE ENTRENAMIENTO
      conf_matrix =confusion_matrix(y_test, y_pred_test)
      confusion_matrix(y_test, y_pred_test)

      array([[1419,  133],
             [ 186,  375]])

[321] # View the classification report for test data and predictions
      print(classification_report(y_test, y_pred_test))

                    precision    recall  f1-score   support

                 0       0.88      0.91      0.90      1552
                 1       0.74      0.67      0.70       561

          accuracy                           0.85      2113
         macro avg       0.81      0.79      0.80      2113
      weighted avg       0.85      0.85      0.85      2113
```

Figura 56: Resultados Random Forest Classifier. Autor: Blanca Mejía, 2024.

3.5.4.2. Resultados obtenidos en el modelo de Árbol de decisión

- **Precisión:**

  Si Cancela (1) el porcentaje de acierto es 0.74%.
  No Cancela (0) el porcentaje de acierto es 0.88%.
  El modelo tiene una precisión es del 74%. Esto indica que, de todas las instancias predichas como positivas, el 74% realmente son positivas. Una precisión relativamente alta sugiere que cuando el modelo predice una clase, es probable que sea correcta.

- ***Recall:***

  Si Cancela (1) el porcentaje de acierto es 0.67%.
  No Cancela (0) el porcentaje de acierto es 0.91%.
  El modelo tiene una recuperación del 67%. Esto significa que el modelo captura el 67% de todas las instancias positivas reales. Una recuperación moderada indica que el modelo no identifica todas las instancias positivas, pero captura más de la mitad.

- ***F1-score:***

  Si Cancela (1) el porcentaje de acierto es 0.70%.
  No Cancela (0) el porcentaje de acierto es 0.90%.
  El modelo tiene una puntuación F1 del 70%. Un valor relativamente alto sugiere que el modelo está encontrando un buen compromiso entre la precisión y la recuperación, en este modelo supera el 50%.
- ***Accuracy:*** 0.85%

  Dada la naturaleza dicotómica de la variable objetivo Churn, el modelo de árbol de

decisión indica un 85%.

En el modelo tiene una exactitud del 85%. Esto significa que aproximadamente el 85% de todas las predicciones (positivas y negativas) son correctas.

- **Resumen**

  La exactitud indica que el modelo está acertando en aproximadamente el 85% de las instancias, lo cual es bastante razonable.

- La precisión y la recuperación con aproximadamente 74% indica que son moderadas, que el modelo tiene un equilibrio aceptable entre identificar instancias positivas y evitar falsos positivos.

- La puntuación F1 sugiere un buen equilibrio, aproximadamente 70% entre precisión y recuperación.

  En conclusión, el modelo tiene un rendimiento generalmente aceptable, el cual deberá ser acoplado al alcance del problema que se desea solucionar con el modelo predictivo para conocer los clientes que cancelarán el servicio.

3.5.4.3. CURVA ROC:

En la **Figura 57** se observa la curva ROC y áreas bajo la curva, en este modelo indica que el modelo puede distinguir en un 89% de capacidad predictiva.

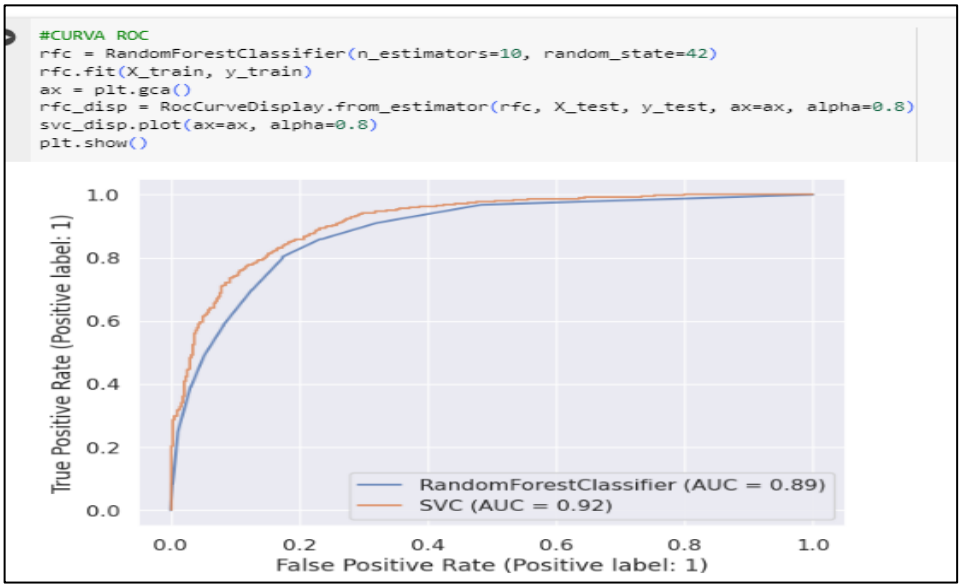


Figura 57: Curva ROC Random Forest Classifier. Autor: Blanca Mejía, 2024.

3.5.5. Modelo Lazy Classifier

- En la **Figura 58** inicialmente se declara la librería *LightGBM* para entrenar un modelo de clasificación.
- Se crea una instancia de *LGBMClassifier*, que es un clasificador basado en el algoritmo de *LightGBM.*
- Se ajusta el clasificador a los datos de entrenamiento *X_train* y *y_train*.
- Se realizan las predicciones sobre las variables de entrenamiento.
- Finalmente imprime la predicción.

```
[324] from lightgbm import LGBMClassifier
      # Create the LGBMClassifier
      lgbm_classifier = LGBMClassifier()
      # Fit the classifier to the training data
      lgbm_classifier.fit(X_train, y_train)
      # Make predictions on the test data
      y_predL = lgbm_classifier.predict(X_test)
      # Evaluate the accuracy of the classifier
      accuracy = accuracy_score(y_test, y_predL)
      print("Accuracy:", accuracy)

      [LightGBM] [Warning] Found whitespace in feature_names, replace with underlines
      [LightGBM] [Info] Number of positive: 1308, number of negative: 3622
      [LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.001598 seconds.
      You can set `force_row_wise=true` to remove the overhead.
      And if memory is not enough, you can set `force_col_wise=true`.
      [LightGBM] [Info] Total Bins 695
      [LightGBM] [Info] Number of data points in the train set: 4930, number of used features: 36
      [LightGBM] [Info] [binary:BoostFromScore]: pavg=0.265314 -> initscore=-1.018527
      [LightGBM] [Info] Start training from score -1.018527
      Accuracy: 0.8580217699952674
```

Figura 58: Construcción Lazy Classifier. Autor: Blanca Mejía, 2024.

3.5.5.1. Matriz de confusión

En la **Figura 59** se construye la matriz de confusión para evaluar los resultados del modelo *Lazy Classifier.*

```
# Confusion matrix
cm_algorithm1 = confusion_matrix(y_test, y_predL)
plt.figure(figsize=(8, 6))
sns.heatmap(cm_algorithm1, annot=True, fmt=".0f", cmap='Blues')
plt.title('Confusion Matrix - Lazy Classifier')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
```

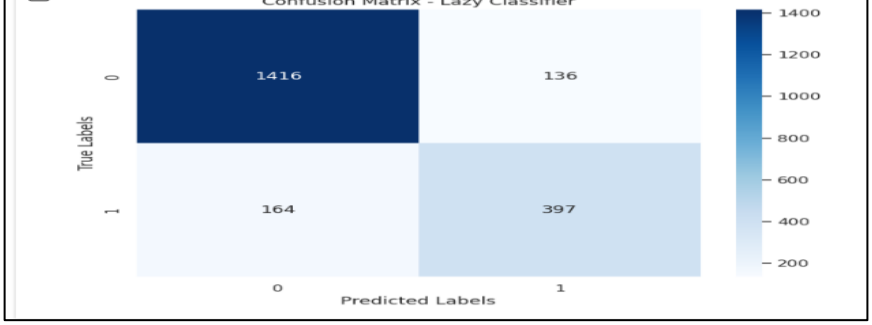


Figura 59: Matriz de confusión Lazy Classifier. Autor: Blanca Mejía, 2024.

En la **Figura 59** se puede observar la matriz de confusión que indica que los valores de la diagonal principal 1416 y 397 corresponden con los valores estimados de forma correcta por el modelo ya que los valores más altos deben ser los de la diagonal principal, en este caso si cumple. Las variables de tipo dicotómico (Yes/No) lo que significa que los valores asignados serán 0 y 1 (0 para No y 1 para Si).

- Si cancela (1).
- No cancela (0).

Verdaderos Positivos (TP): 1416.
Verdaderos Negativos (TN): 397.
Falsos Positivos (FN): 136.
Falsos Negativos (FP): 164.

En la **Figura 60** se puede observar los resultados obtenidos del modelo *Lazy Classifier* que se detallarán uno a uno más adelante.

```
from lightgbm import LGBMClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, f1_score
lgbm_classifier = LGBMClassifier()
lgbm_classifier.fit(X_train, y_train)
y_predL = lgbm_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_predL)
print("Accuracy:", accuracy)
conf_matrix = confusion_matrix(y_test, y_predL)
print("Confusion Matrix:\n", conf_matrix)
class_report = classification_report(y_test, y_predL)
print("Classification Report:\n", class_report)
f1 = f1_score(y_test, y_predL)
print("F1 Score:", f1)
```

```
[LightGBM] [Warning] Found whitespace in feature_names, replace with underlines
[LightGBM] [Info] Number of positive: 1308, number of negative: 3622
[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.002165 seconds.
You can set `force_row_wise=true` to remove the overhead.
And if memory is not enough, you can set `force_col_wise=true`.
[LightGBM] [Info] Total Bins 695
[LightGBM] [Info] Number of data points in the train set: 4930, number of used features: 36
[LightGBM] [Info] [binary:BoostFromScore]: pavg=0.265314 -> initscore=-1.018527
[LightGBM] [Info] Start training from score -1.018527
Accuracy: 0.8580217699952674
Confusion Matrix:
 [[1416  136]
 [ 164  397]]
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.91      0.90      1552
           1       0.74      0.71      0.73       561

    accuracy                           0.86      2113
   macro avg       0.82      0.81      0.81      2113
weighted avg       0.86      0.86      0.86      2113
```

Figura 60: Resultados Lazy Classifier. Autor: Blanca Mejía, 2024

3.5.5.2. Resultados obtenidos en el modelo Lazy Classifier

- **Precisión:**

  Si Cancela (1) el porcentaje de acierto es 0.74%.
  No Cancela (0) el porcentaje de acierto es 0.90%.
  El modelo tiene una precisión es del 74%. Esto indica que, de todas las instancias predichas como positivas, el 74% realmente son positivas. Una precisión relativamente alta sugiere que cuando el modelo predice una clase, es probable que sea correcta.

- ***Recall:***

  Si Cancela (1) el porcentaje de acierto es 0.71%.
  No Cancela (0) el porcentaje de acierto es 0.91%.
  El modelo tiene una recuperación del 71%. Esto significa que el modelo captura el 71% de todas las instancias positivas reales. Una recuperación moderada indica que el modelo no identifica todas las instancias positivas, pero captura más de la mitad.

- ***F1-score:***

  Si Cancela (1) el porcentaje de acierto es 0.73%.
  No Cancela (0) el porcentaje de acierto es 0.90%.
  El modelo tiene una puntuación F1 del 73%. Un valor relativamente alto sugiere que el modelo está encontrando un buen compromiso entre la precisión y la recuperación, en este modelo supera el 50%.
- ***Accuracy:*** 0.86%

  Dada la naturaleza dicotómica de la variable objetivo Churn, el modelo de árbol de decisión indica un 85%.
  En el modelo tiene una exactitud del 85%. Esto significa que aproximadamente el 86% de todas las predicciones (positivas y negativas) son correctas.

- **Resumen:**

  La exactitud indica que el modelo está acertando en aproximadamente el 86% de

las instancias, lo cual es bastante razonable.

La precisión y la recuperación con aproximadamente 74% indica que son moderadas, que el modelo tiene un equilibrio aceptable entre identificar instancias positivas y evitar falsos positivos.

La puntuación F1 sugiere un buen equilibrio, aproximadamente 74% entre precisión y recuperación.

En conclusión, el modelo tiene un rendimiento generalmente aceptable, el cual deberá ser acoplado al alcance del problema que se desea solucionar con el modelo predictivo para conocer los clientes que cancelarán el servicio.

3.5.5.3. CURVA ROC

```
from sklearn.metrics import roc_curve, roc_auc_score
# Calcular las probabilidades de las clases
y_pred_prob = model.predict_proba(X_test)
# Calcular la tasa de verdaderos positivos, tasa de falsos positivos y umbrales usando roc_curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob[:,1])
# Calcular el área bajo la curva ROC
roc_auc = roc_auc_score(y_test, y_pred_prob[:,1])
# Graficar la curva ROC
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='red', label='ROC curve (AUC = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend()
plt.show()
```

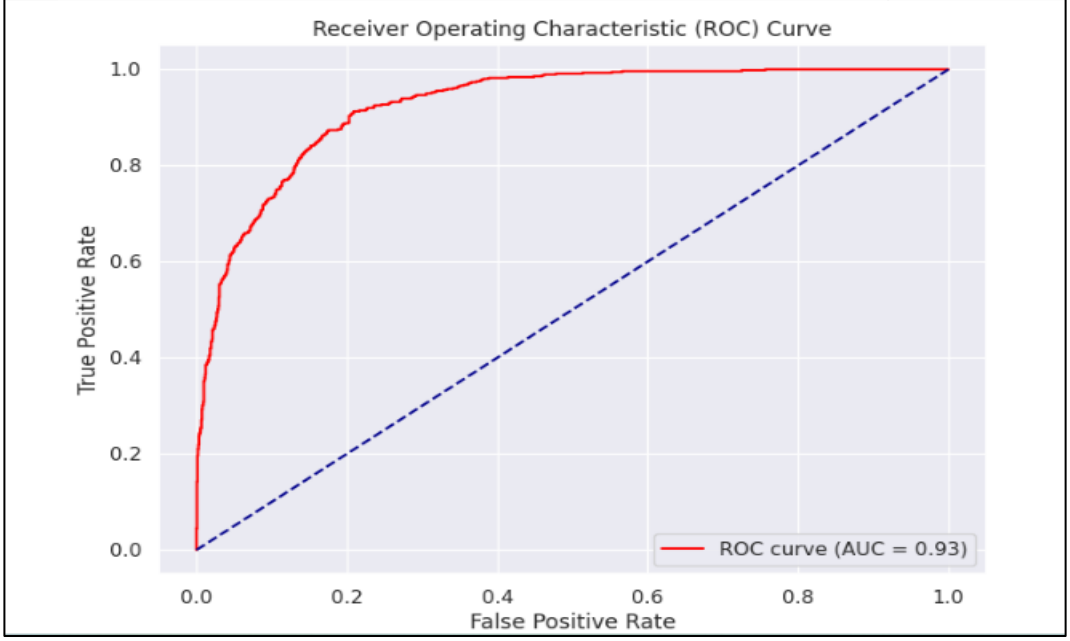


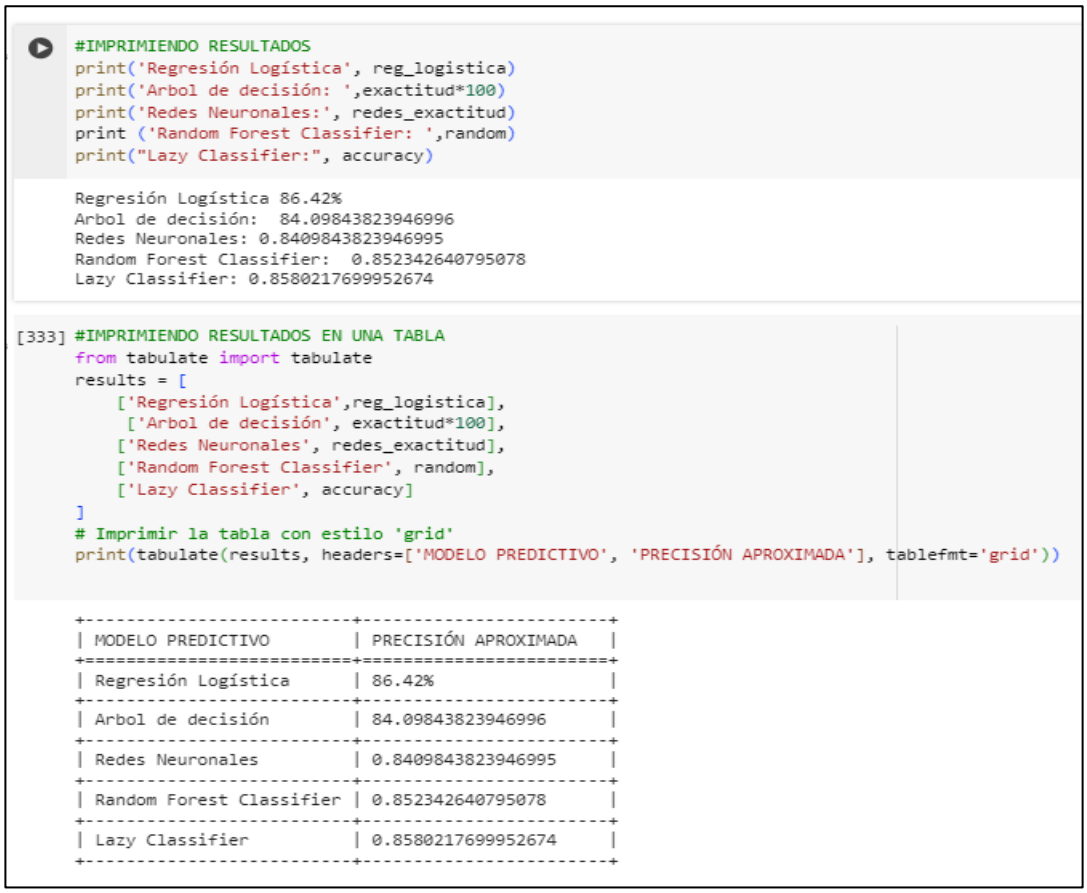
Figura 61: Curva ROC Lazy Classifier. Autor: Blanca Mejía, 2024.

En la **Figura 61** se observa la curva ROC y áreas bajo la curva, en este modelo indica que el modelo puede distinguir en un 80% de capacidad predictiva.

3.6. Evaluación General

En la **Figura 62** se realiza una tabla de comparación entre los resultados de los diferentes modelos aplicados en este Proyecto, en la imagen detalla la exactitud de todos los modelos.

El modelo que mayor exactitud tiene es el de Regresión Logística con un 86%, se puede indicar además que la diferencia con los otros modelos es mínima, únicamente entre un 1% y 2% lo que indica que todos los modelos tienen una exactitud aceptable y que cumplen con los objetivos del negocio indicados inicialmente. Para este proyecto se seleccionará el Modelo de Regresión Logística, como el más preciso.



```
#IMPRIMIENDO RESULTADOS
print('Regresión Logística', reg_logistica)
print('Arbol de decisión: ',exactitud*100)
print('Redes Neuronales:', redes_exactitud)
print ('Random Forest Classifier: ',random)
print("Lazy Classifier:", accuracy)
```

```
Regresión Logística 86.42%
Arbol de decisión:  84.09843823946996
Redes Neuronales: 0.8409843823946995
Random Forest Classifier:  0.852342640795078
Lazy Classifier: 0.8580217699952674
```

```
[333] #IMPRIMIENDO RESULTADOS EN UNA TABLA
from tabulate import tabulate
results = [
    ['Regresión Logística',reg_logistica],
     ['Arbol de decisión', exactitud*100],
    ['Redes Neuronales', redes_exactitud],
    ['Random Forest Classifier', random],
    ['Lazy Classifier', accuracy]
]
# Imprimir la tabla con estilo 'grid'
print(tabulate(results, headers=['MODELO PREDICTIVO', 'PRECISIÓN APROXIMADA'], tablefmt='grid'))
```

```
+--------------------------+------------------------+
| MODELO PREDICTIVO        | PRECISIÓN APROXIMADA   |
+==========================+========================+
| Regresión Logística      | 86.42%                 |
+--------------------------+------------------------+
| Arbol de decisión        | 84.09843823946996      |
+--------------------------+------------------------+
| Redes Neuronales         | 0.8409843823946995     |
+--------------------------+------------------------+
| Random Forest Classifier | 0.852342640795078      |
+--------------------------+------------------------+
| Lazy Classifier          | 0.8580217699952674     |
+--------------------------+------------------------+
```

Figura 62: Evaluación general. Autor: Blanca Mejía, 2024.

En la **Figura 63** se realiza la codificación para obtener la curva ROC de todos los modelos desarrollados en este Proyecto.

```python
from sklearn.metrics import f1_score, recall_score, precision_score, roc_curve, auc, roc_auc_score
from sklearn.metrics import f1_score, recall_score, precision_score, roc_curve, auc
import pandas as pd
# Diccionarios para almacenar las métricas
f1_scores = {}
recall_scores = {}
precision_scores = {}
roc_auc_scores = {}

# Preparando la figura para la curva ROC
plt.figure(figsize=(10, 6))

# Entrenar, evaluar y calcular métricas para cada modelo
for i, (model, name) in enumerate(zip(models, model_names)):
    # Haciendo predicciones con el modelo
    y_pred = model.predict(X_test)

    # Calculando métricas
    f1_scores[name] = f1_score(y_test, y_pred)
    recall_scores[name] = recall_score(y_test, y_pred)
    precision_scores[name] = precision_score(y_test, y_pred)

    # Calculando la curva ROC y el área bajo la curva si el modelo soporta probabilidades
    if hasattr(model, "predict_proba"):
        y_pred_proba = model.predict_proba(X_test)[:, 1]
        fpr, tpr, _ = roc_curve(y_test, y_pred_proba)
        roc_auc = auc(fpr, tpr)
        roc_auc_scores[name] = roc_auc

        # Agregar la curva ROC al gráfico
        plt.plot(fpr, tpr, label=f'{name} (AUC = {roc_auc:.2f})')

# Finalizando la gráfica de la curva ROC
plt.plot([0, 1], [0, 1], 'k--', label='Random Chance')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curves')
plt.legend(loc='lower right')
plt.show()

f1_scores_table = pd.DataFrame(f1_scores.items(), columns=['Model', 'F1 Score'])
recall_scores_table = pd.DataFrame(recall_scores.items(), columns=['Model', 'Recall Score'])
precision_scores_table = pd.DataFrame(precision_scores.items(), columns=['Model', 'Precision Score'])

# Función para imprimir tablas con bordes utilizando tabulate
def print_table_with_border(df, title):
    print(f"{'=' * 10} {title} {'=' * 10}")
    print(tabulate(df, headers='keys', tablefmt='pretty'))
```

Figura 63: Curva ROC general. Autor: Blanca Mejía, 2024.

3.6.1. CURVA ROC

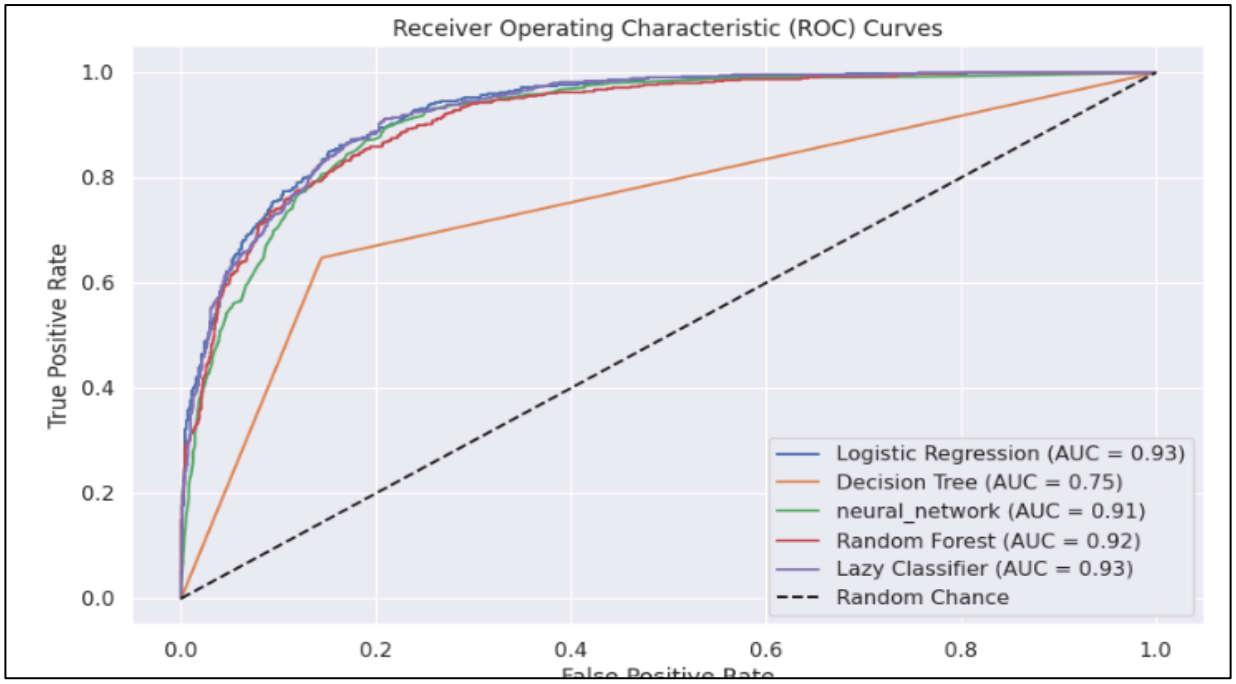


Figura 64: Curva ROC general. Autor: Blanca Mejía, 2024.

En la **Figura 64** se puede apreciar la precisión bajo la curva de todos los modelos:

- La exactitud del modelo de Regresión Logística es de 93%.
- La exactitud del modelo de Árbol de decisión es de 75%.
- La exactitud del modelo de Redes Neuronales es de 91%.
- La exactitud del modelo de *Random Forest Classifier* es de 92%.
- La exactitud del modelo *Lazy Classifier* es de 93%.

3.6. Despliegue

En la **Figura 65** se detalla la fase de Despliegue. Esta es la última fase de la metodología CRISP-DM y su objetivo es el de explicar al cliente como poner en funcionamiento el proyecto que se ha construido en las fases anteriores, así como exponerle los resultados obtenidos de manera que lo pueda entender fácilmente. Otro de sus objetivos es la de crear una estrategia para el mantenimiento del proyecto y producir un informe en el que se incluyan posibles mejoras para el futuro y un listado de las dificultades encontradas a la hora de realizarlo.

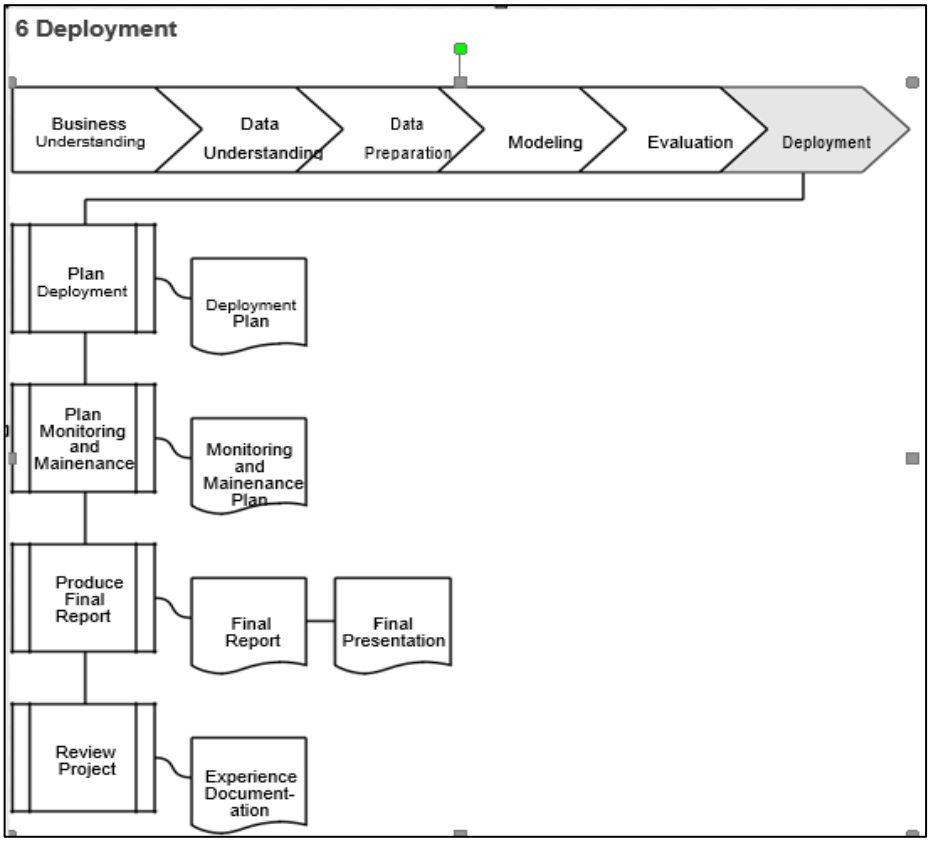


Figura 65: Fase Despliegue. Autor: Desconocido, recuperado 2024.

a. Planificar la implementación

Para poder implementar este proyecto en el negocio real sería necesario en primer lugar tener acceso a la base de datos real del negocio, es decir la base de datos que contiene toda la información relativa a los clientes de la empresa. A partir de ahí, los pasos a seguir serían los mismos que se han seguido en este documento desde la comprensión del negocio hasta la implantación. Si bien, cabe mencionar que existirán algunas fases, como la de comprensión y preparación de los datos, que en el negocio real probablemente sean más complejas y llevarán más tiempo en el desarrollo del proyecto ya que se puede esperar que en la base de datos real se tengan muchos más registros y estos mismos contengan más ruido que en nuestra base de datos muestral utilizada específicamente para este uso.

En segundo lugar, sería necesario que en el negocio (empresa de telecomunicaciones) se use una base de datos robusta, de no ser así se tendrían dos opciones, la primera sería exportar la base de datos actual a una base de datos más robusta, y la segunda sería utilizar algún software de minería de datos que mejor se adapte a la base de datos original, para esto sería necesario hacer un estudio previo que determine que herramienta es la más apropiada.

b. Planear Monitoreo y Mantenimiento

La supervisión y mantenimiento de la implementación de este proyecto es una fase importante de la metodología CRISPDM, debido a que los datos que se procesan con mucha frecuencia pueden ser modificados por el personal de la empresa de telecomunicaciones. Los datos pueden ser modificados por diferentes motivos como haber realizado una codificación incorrecta, haber registrado una novedad de la mesa de ayuda, activaciones y desactivaciones de servicio, etc.

El volumen de estos datos en movimiento es grande motivo por el cual la extracción de las muestras debe ser realizada cuidadosamente y realizando siempre *backups* de los datos explotados en cada proceso. La minería de datos debería ser realizada en periodos de tres meses (trimestres) ya que esta es la medida de tiempo utilizada en la empresa para realizar los análisis de los comportamientos de los clientes en cuanto a los servicios contratados, sin embargo, esta medida podría variar en cualquier momento en función del plan comercial que esté vigente en cada momento.

Como plan de supervisión y mantenimiento se podría establecer los siguientes procesos:

- Extracción y almacenamiento trimestral de los datos guardando la información

obtenida en formato de hoja de cálculo.

- Distribución de los datos en función de los modelos de software de minería de datos a trabajar.

- Los archivos de la explotación de datos deberán ser guardados en soporte magnético en la propia empresa, almacenándolos por ejemplo en carpetas ordenadas por procesos trimestrales.

- Los resultados obtenidos en cada explotación de datos deberán ser llevados a formato de hoja de cálculo y generar gráficas de distintos tipos para una mejor visualización e interpretación de los resultados obtenidos en cada periodo.

c. Producir informe final

En este paso se debe presentar un informe resumiendo los puntos importantes del proyecto y la experiencia adquirida durante su desarrollo. El público al que va dirigido este informe sería el personal de la empresa de telecomunicaciones encargado del seguimiento a los clientes (control de calidad, postventa, servicio al cliente, etc.), de tal manera que se pueda estudiar la situación actual y tomar medidas correctivas para la mejora de los servicios ofrecidos.

El uso de la metodología CRISP-DM en este proyecto ha permitido encontrar un comportamiento predictivo a la hora de estimar la permanencia de los clientes en los servicios contratados. Se ha podido encontrar un plan de extracción, normalización, y codificación de datos para la realización de procesos de minería de datos trimestrales.

d. Repasando las diferentes etapas que se ha seguido para llegar al objetivo

- La primera etapa ha sido la de conseguir un entendimiento del negocio para tener claro los objetivos del negocio delimitando claramente el objetivo final que se logra alcanzar, en este caso detectar y predecir la tasa de cancelación de los servicios y que particularidades de los contratos y comportamiento del cliente censado a través de la mesa de ayuda están incidiendo en la cancelación de los contratos.

- En base al *dataset* se procedió a realizar un entendimiento de los datos y sacar más conclusiones al margen de los objetivos iniciales de la minería. Los datos fueron preparados y formateados para que sean de fácil aplicación para el modelo de predicción seleccionado para atender el problema.

- Se realizó la elección de las técnicas de modelado y la ejecución de dichas técnicas sobre los datos empleando la herramienta escogida para ello (*Python*). Esta herramienta facilitó por completo la aplicación de los modelos ya que nos permitió ver de manera muy intuitiva y visual cuales eran las técnicas más adecuadas para nuestra base de datos. Una vez obtenidos los modelos, se analizaron para determinar la adecuación o no de los mismos. En este caso determinamos que las métricas de evaluación sobre los resultados del modelo de regresión logística eran lo suficientemente fiables.

e. Revisar el proyecto (Experiencias durante el desarrollo)

En esta última etapa de la metodología se debe hacer una evaluación de aquellas cosas que se hicieron correctamente y aquellas que no, así como posibles mejoras para que en las futuras ejecuciones de la minería de datos se vayan corrigiendo los errores y se obtengan mejores resultados.

Lo ideal sería conveniente tener acceso a la base de datos para evaluar otros períodos o set de datos para ampliar la visión del problema planteado de la deserción. El tratamiento de estos nuevos conjuntos de datos permitiría incrementar aún más la fiabilidad de los modelos de minería de datos elegidos en este proyecto. Esto se puede interpretar como algo positivo ya que, si se ha dado por válido el modelo seleccionado y sin disponer de una mayor cantidad de datos que reflejen la realidad del negocio, los resultados de un *dataset* real se pueden esperar resultados más precisos con acceso a una mayor cantidad de datos.

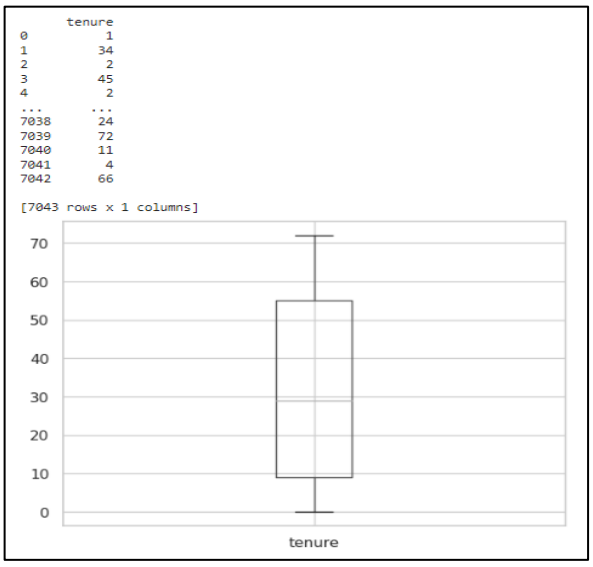


Figura 66: Duración contrato. Autor: Blanca Mejía, 2024.

En la **Figura 66** se puede observar que los clientes en su mayoría eligieron servicios para 29 meses.

En la **Figura 67** se observa la relación de meses por género, es decir cuantos meses aproximadamente contratan las mujeres vs. los hombres. Se puede apreciar que las mujeres contratan 32.24 meses, mientras que los hombres 32.5, realmente es la misma cantidad de meses. Están equilibrados los valores.

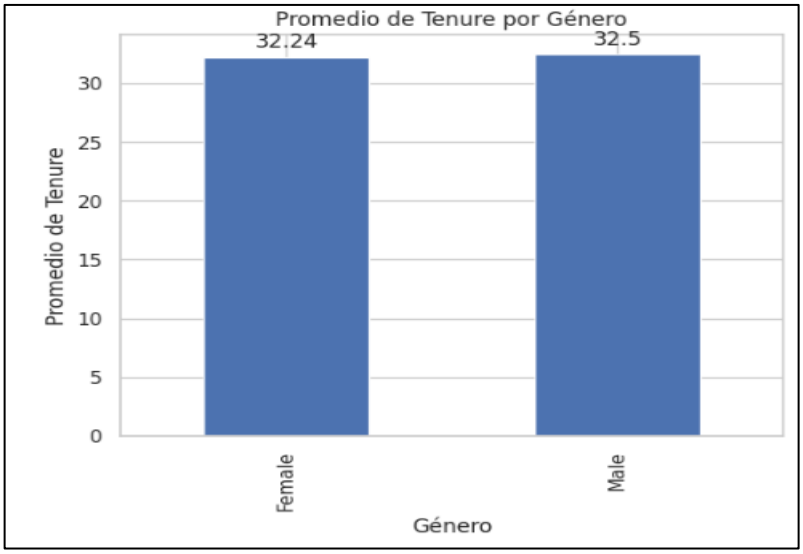


Figura 67: Promedio contrato por género. Autor: Blanca Mejía, 2024.

En la **Figura 68** se puede apreciar, que los clientes que son adultos mayores utilizan en promedio, por más tiempo los servicios. Yes = Si es cliente adulto mayor, No = No es cliente adulto mayor.

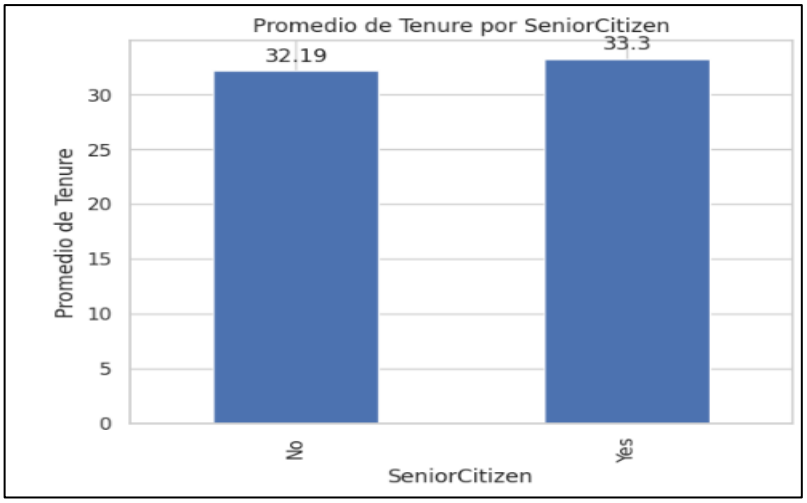


Figura 68: Promedio contrato por tercera edad. Autor: Blanca Mejía, 2024.

En la **Figura 69** se puede apreciar la duración del contrato en porcentajes, siendo el tipo mes a mes el más contratado.

Duración Contrato:

Month-to-month 3875 = 55%

Two year 1695 = 24.1%

One year 1473 = 20.9%

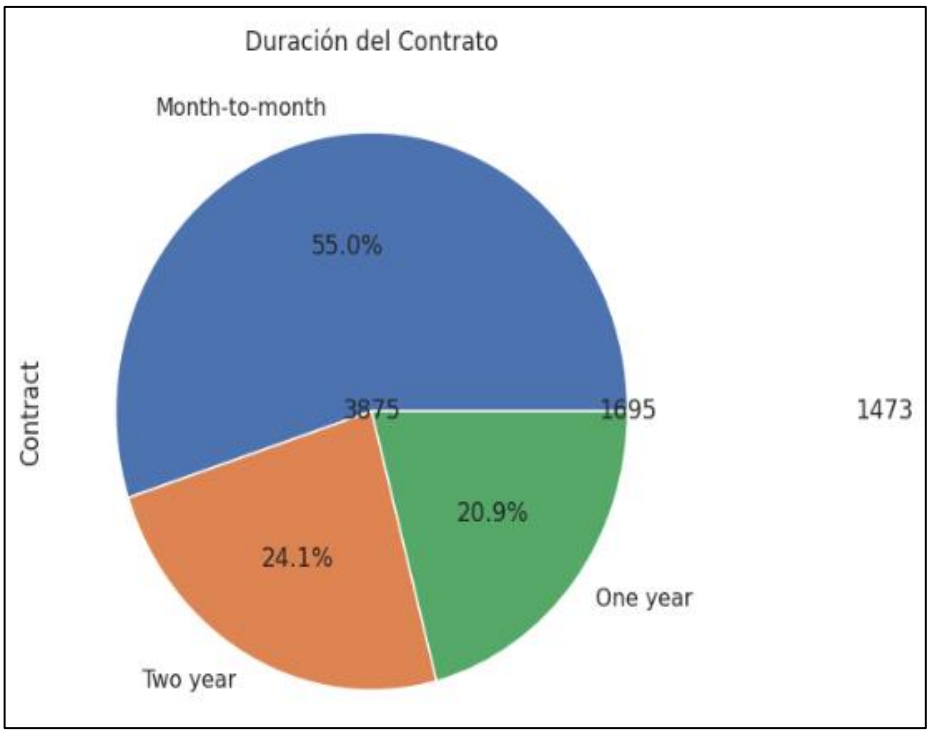


Figura 69: Tiempo de duración del contrato. Autor: Blanca Mejía, 2024.

En la **Figura 70** se puede observar, analizamos el promedio de los cargos mensuales asignados a los clientes en función del tipo de contrato que han suscrito.

Cargos mensuales:

Month-to-month 66.39%

One year 65.04%

Two year 60.77%

Siendo el periodo de contrato *Month-to-month* el que más contrataron los clientes.

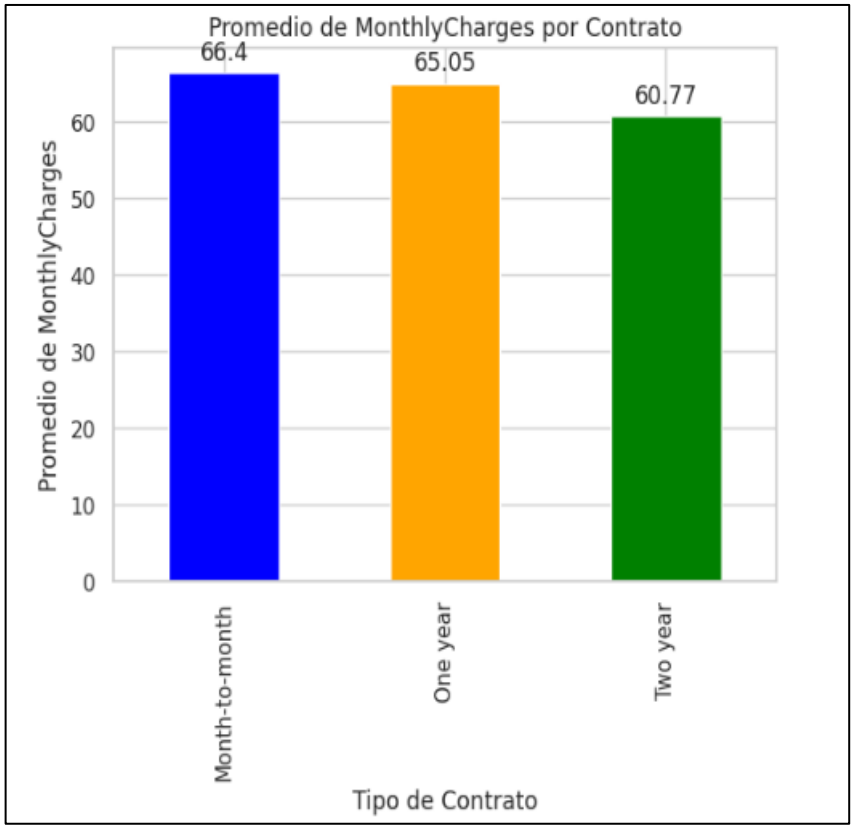


Figura 70: Promedio de los cargos mensuales. Autor: Blanca Mejía, 2024.

En la **Figura 71** se puede observar la información general de las variables de los clientes vs. la variable objetivo *Churn*.

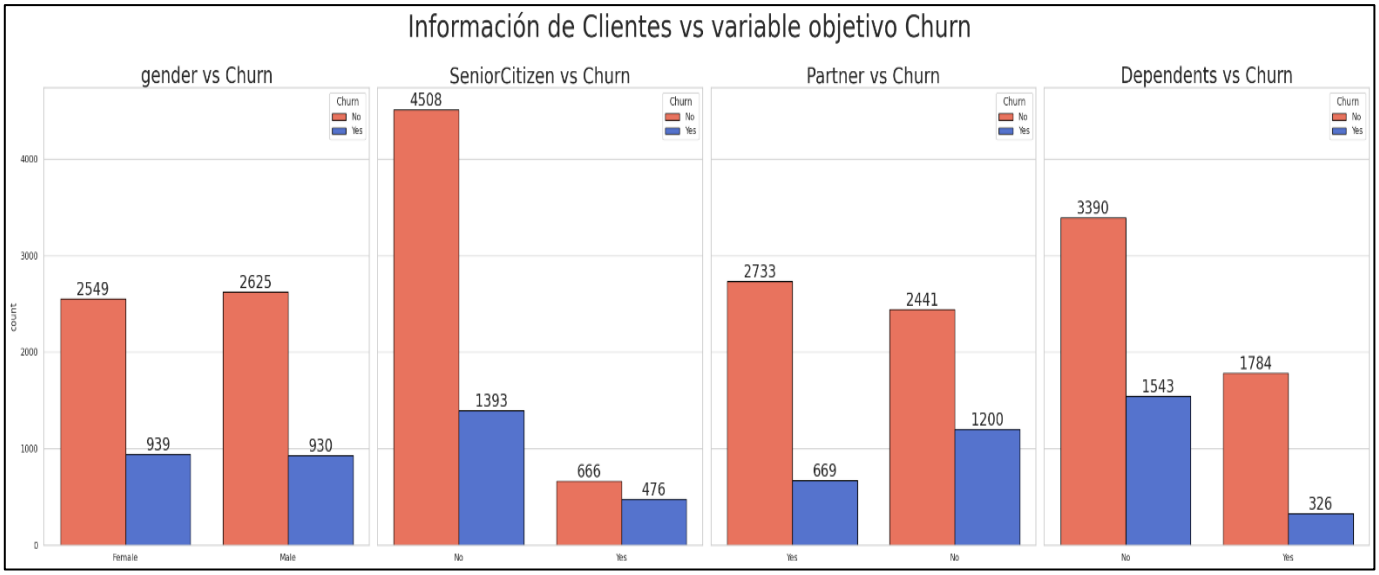


Figura 71: Información general de clientes vs. Churn. Autor: Blanca Mejía, 2024.

		CHURN	
		SI CANCELA EL SERVICIO	NO CANCELA EL SERVICIO
GENDER	FEMENINO	939	2549
	MASCULINO	930	2625
TERCERA EDAD	SI	1393	4508
	NO	476	666
TIENE PAREJA	SI	669	2733
	NO	1200	2441
DEPENDIENTES	SI	326	1784
	NO	1543	3309

Tabla 2: Clientes vs. Churn. Autor: Blanca Mejía, 2024.

En la **Figura 72** se puede observar la información de los servicios vs. la variable objetivo *Churn*. En la **Tabla 4**, se observa detalladamente la cantidad de clientes por servicios contratados que cancelan o no el servicio.

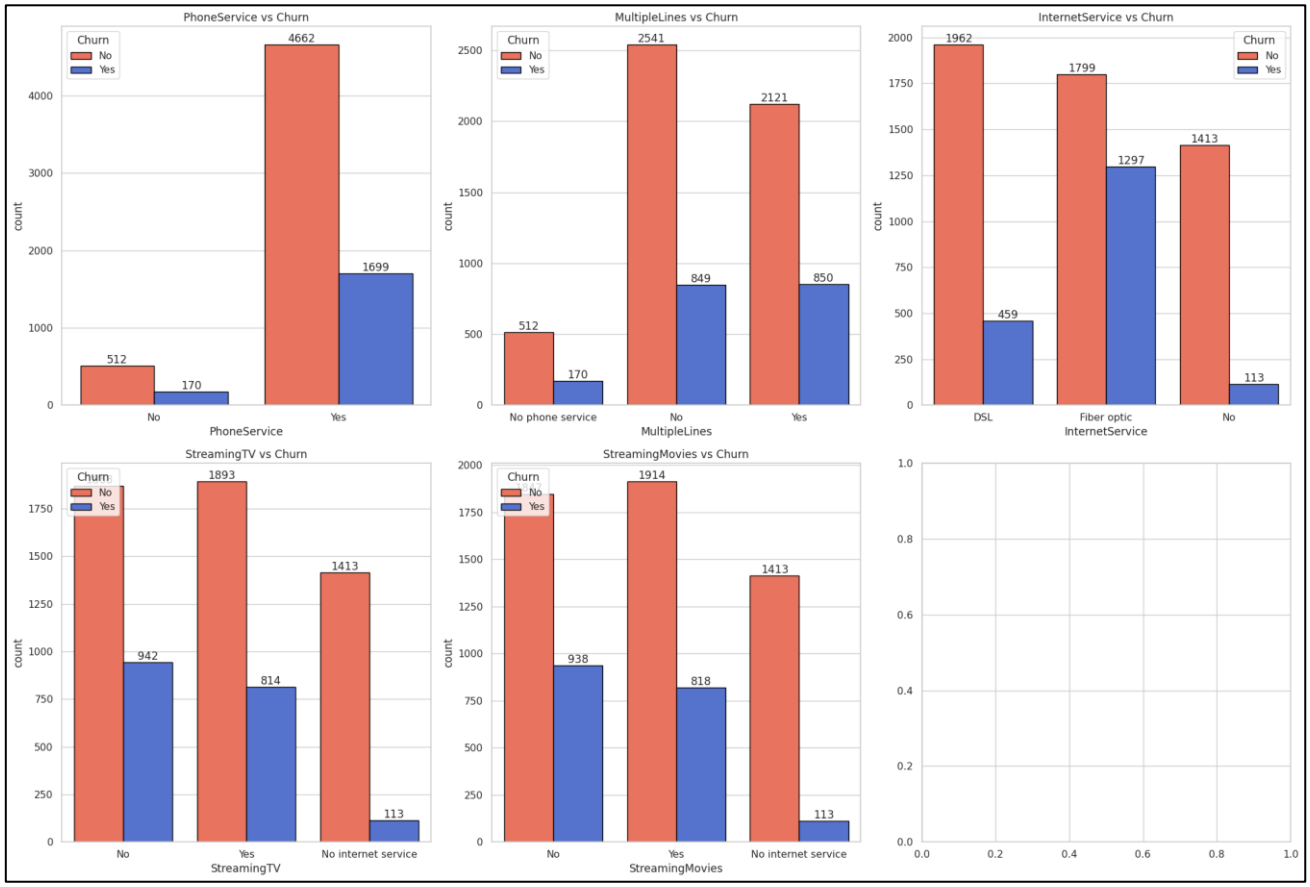


Figura 72:Información general de servicios vs. Churn. Autor: Blanca Mejía, 2024.

		CHURN	
		SI CANCELA EL SERVICIO	NO CANCELA EL SERVICIO
PHONE SERVICE	SI	1699	4662
	NO	170	512
MULTIPLELINES	SI	850	2121
	NO PHONE SERVICE	170	512
	NO MULTIPLELINES	849	2541
INTERNET SERVICE	DSL	459	1962
	NO	113	1413
	FIBRA OPTICA	1297	1799
STREAMING TV	SI	814	1893
	NO	942	1868
	NO INTERNET SERVICE	113	1413
STREAMING MOVIES	SI	818	1914
	NO	938	1847
	NO INTERNET SERVICE	113	1413

Tabla 3: Servicios vs. Churn. Autor: Blanca Mejía, 2024.

En la **Figura 73** se puede observar la información de las formas de pago vs. la variable objetivo *Churn*. En la **Tabla 5**, se observa detalladamente la cantidad de clientes que cancelan o no el servicio según la forma de pago.

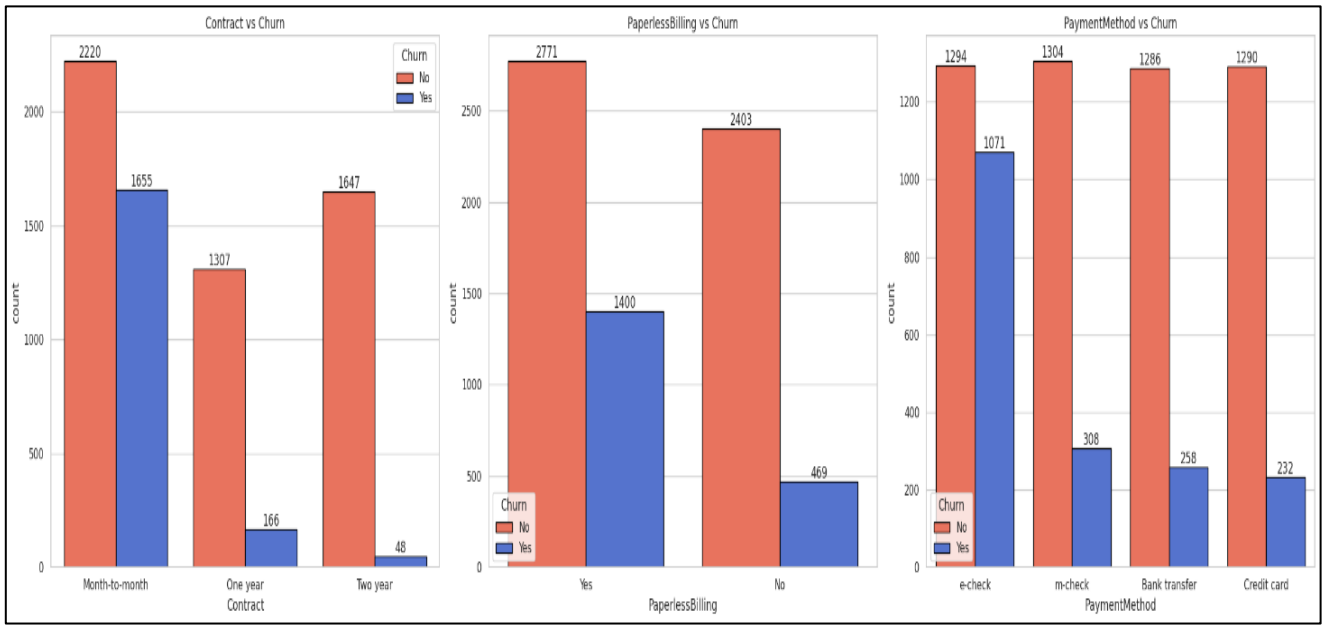


Figura 73: Información formas de pago vs. Churn. Autor: Blanca Mejía, 2024.

		CHURN	
		SI CANCELA EL SERVICIO	NO CANCELA EL SERVICIO
CONTRACT	MONTH TO MONTH	1655	2220
	ONE YEAR CONTRACT	166	1307
	TWO YEARS	48	1647
PAPERLESSBILLING	SI	1400	2771
	NO	469	2403
PAYMENTMETHOD	E-CHECK	1071	1294
	M-CHECK	308	1304
	BANK TRANSFER	258	1286
	CREDIT CARD	232	1290

Tabla 4: Formas de pago vs. Churn. Autor: Blanca Mejía, 2024.

En la **Figura 74** se puede observar la información de los tipos de soporte de servicios que ha contratado el cliente vs. la variable objetivo *Churn*. En la **Tabla 6**, se observa detalladamente la cantidad de clientes que cancelan o no el servicio según el tipo de soporte de servicio que reciben.

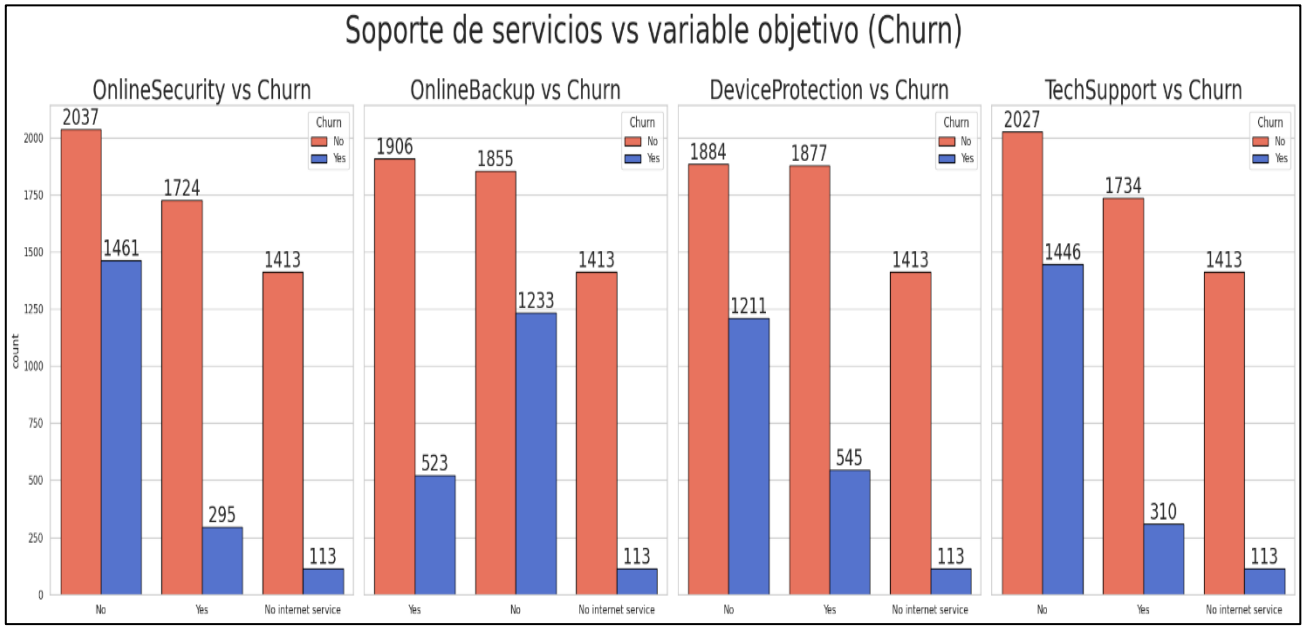


Figura 74: Información tipos de soporte vs. Churn. Autor: Blanca Mejía, 2024.

		CHURN	
		SI CANCELA EL SERVICIO	NO CANCELA EL SERVICIO
ONLINESECURITY	SI	295	1724
	NO	1461	203
	NO INTERNET SERVICE	113	1413
ONLINEBACKUP	SI	523	1906
	NO	1233	1855
	NO INTERNET SERVICE	113	1413
DEVICEPROTECTION	SI	545	1877
	NO	1211	1884
	NO INTERNET SERVICE	113	1413
TECHSUPPORT	SI	310	1734
	NO	1446	2027
	NO INTERNET SERVICE	113	1413

Tabla 5: Tipos de soporte vs. Churn. Autor: Blanca Mejía, 2024.

4. Conclusiones y Recomendaciones

4.1 Conclusiones

- La implementación de modelos de predicción mediante aprendizaje automático para anticipar el comportamiento de los clientes y prever solicitudes de cancelación de servicios en la empresa de Telecomunicaciones es una estrategia eficaz y proactiva para identificar el riesgo e índice de abandono de clientes y que la empresa pueda actuar de manera oportuna antes que se convierta en una situación crítica e incontrolable. A través del comportamiento de los clientes y en base al índice de cancelaciones la empresa es responsable en implementar técnicas, estrategias y mejorar la atención a los clientes. Permitiendo que el cliente sienta mayor satisfacción a largo plazo.

- Al utilizar modelos predictivos, la empresa puede identificar patrones y señales tempranas de insatisfacción, permitiendo la adopción de medidas preventivas y personalizadas para retener a los clientes, lo que resulta en una significativa reducción de la tasa de cancelación de servicios y, en última instancia, en la mejora de la satisfacción del cliente y la retención a largo plazo.

- Al explicar el funcionamiento de los modelos predictivos a través del aprendizaje automático no solo facilita la comprensión interna sobre cómo operan los modelos, sino

que también contribuye a la transparencia y confianza en la aplicación de estas tecnologías. Al respaldar la implementación con principios sólidos, la empresa puede mejorar la aceptación y la toma de decisiones informadas, estableciendo una base sólida para optimizar continuamente los modelos predictivos y su integración en la gestión de la relación con el cliente.

- La aplicación de la metodología CRISP-DM para predecir el motivo detrás de las cancelaciones de servicio en clientes ha demostrado ser una estrategia efectiva y estructurada. Al seguir las fases de CRISP-DM, se ha logrado un enfoque sistemático para el desarrollo de modelos predictivos. Garantiza la calidad y la preparación adecuada de los datos, resultando en modelos más precisos y relevantes. La implementación exitosa de esta metodología brinda a la empresa una visión valiosa sobre los motivos de cancelación, permitiendo la toma de decisiones informadas y estrategias específicas para retener a los clientes de manera más efectiva.

- El diseño de un modelo predictivo basado en quejas de usuarios ha demostrado ser una estrategia efectiva para prever la cancelación de servicios. Este enfoque permite identificar patrones de insatisfacción, facilitando intervenciones personalizadas y mejorando la retención de clientes. La implementación exitosa fortalece la relación con los usuarios al anticipar y abordar problemas, mejorando la satisfacción general del cliente.

- Al comprender el comportamiento y tendencias en un cliente previas a la cancelación, la empresa puede anticipar y abordar proactivamente las preocupaciones de los clientes, mejorando la retención y la satisfacción general del cliente.

4.2 Recomendaciones

- Al implementar modelos predictivos en la empresa es prescindible mantener las bases de datos constantemente actualizadas para identificar de manera oportuna de posibles patrones de cancelación. De igual manera los modelos predictivos ayudan a mantener bajo vigilancia continua el comportamiento de los clientes y en base a ello tomar decisiones inmediatas.

- Todo el personal debe estar actualizado sobre el funcionamiento de los modelos predictivos y la lógica de negocio, comprendiendo sobre cada una de las fases de la metodología CRISP-DM para poder identificar en cada fase los posibles incidentes y

aplicar punto de mejora en cada una de las etapas ya que por su naturaleza de ser un modelo iterativo lo permite.

- Aprovechar el *feedback* y retroalimentación de los clientes que cancelan el servicio para recopilar y analizar las quejas de los usuarios y almacenarlas sobre una base de conocimientos para tener una fuente valiosa para mejorar constantemente los modelos predictivos, abordar problemas específicos, comprender mejor las necesidades y expectativas.

- Desarrollar estrategias de retención personalizadas basadas en los patrones identificados en el comportamiento de los clientes antes de la cancelación e implemente sistemas de atención al cliente mejorados como *chatbots* personalizados más rápidos y eficaces. para abordar rápidamente las preocupaciones identificadas.

- Establecer un sistema de monitoreo continuo para evaluar las nuevas tendencias e interés en el comportamiento de los clientes. En base a los resultados ajustar los modelos predictivos según sea necesario para adaptarse a nuevas tendencias y marketing del mercado.

Bibliografía

Blum, A. (1992). Redes neuronales: un marco orientado a objetos para construir sistemas conexionistas. New York: John Wiley & Sons.

Breiman, L. (2001). Random Forest. California: Statistics Department.

Chapman P., Clinton J., Kerber R., Khabaza T., Reinartz T., Shearer C. y Wirth R. (2000). CRISP-DM 1.0 Step-by-step Data Mining Guide.

Fayyad, U. (1996). Data mining and knowledge discovery: making sense out of data. IEEE Computer Society, 11(5), pag 20–25.

HANSEN, A.; MOURITSEN, J. 1999. " Managerial technology and Netted Networks: Competitiveness in Action - The work of translating performance in a high tech firm, Organization" . Volumen 6, Número 3.

Jiawei Han, Data Mining: Concepts and Techniques Second Edition, University of Illinois at Urbana-Champaign, 2006.

Jiawei Han, Data Mining: Concepts and Techniques Second Edition, University of Illinois at Urbana-Champaign, 2006.

Molero G. y Céspedes Y. (2014). Data Mining and Knowledge Discovery: An Introduction. Capítulo de libro Knowledge Discovery in Databases. Ed. Academy Publish.

P. Zikopoulos and C. Eaton, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, 1st ed. McGraw-Hill Osborne Media, 2011.

Russell, S. J., & Norvig, P. (2010). Artificial Intelligence: A Modern Approach. Prentice Hall.

Russom Philip. (2011).Big Data Analytics. Tdwi Research. 14. 6-8.

Sahu, M. K., Pandey, R., and Silakari, S. Analysis of customer churn prediction in telecom sector usinglogistic regression and decision tree. 62–67.

SAMMUT, C. y WEBB, G.I. Encyclopedia of machine learning. Springer Science & Business Media. 2011.

SAS Institute (1998). Data Mining and the Case for Sampling. Data Mining Using SAS Enterprise Miner.

ANEXO 1

Algoritmos de Aprendizaje automático mediante clasificación aplicando aprendizaje supervisado.

```
# -*- coding: utf-8 -*-

"""PROYECTO FINAL LUCIA MEJIA.ipynb

Automatically generated by Colaboratory.

Original file is located at

    https://colab.research.google.com/drive/1TqFGH9KNQKdyB_FMweISI4_EC5fsKtI3

PROYECTO DE TITULACIÓN

TEMA: MODELO PREDICTIVO DE FIDELIZACIÓN DE CLIENTES.

Para realizar el modelo predictivo se aplicará la metodologia CRISP-DM.

1.-ENTENDIMIENTO DEL NEGOCIO

Mediante el desarrollo del presente modelo predictivo se desea realizar un

análisis de los clientes que cancelarán el servicio en una empresa de Telecomunicaciones.

# Importe de Librerias

"""



# Commented out IPython magic to ensure Python compatibility.

# Librerías

import numpy as np

import pandas as pd

from pandas import read_csv
```

```
from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report

from sklearn.metrics import confusion_matrix

from sklearn.metrics import accuracy_score

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

from sklearn import metrics

from sklearn.metrics import mean_squared_error, r2_score

from sklearn.datasets import load_wine

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import RocCurveDisplay

from sklearn.model_selection import train_test_split

from sklearn.svm import SVC

import missingno as msno

import plotly.express as px

import plotly.graph_objects as go

from plotly.subplots import make_subplots

from sklearn.metrics import precision_score

from sklearn.metrics import recall_score
```

```
from sklearn.metrics import f1_score

from sklearn.metrics import confusion_matrix

from sklearn.metrics import roc_curve, auc

from sklearn.preprocessing import StandardScaler

from sklearn import linear_model


# %matplotlib inline


"""# Carga del Dataset"""


#vincular a google drive

from google.colab import drive

drive.mount('/content/drive/')


#Ruta donde se encuentra el dataset

ruta = "/content/drive/My Drive/CIENCIA DE DATOS/Data Telecomunicaciones.csv"

Clientes = pd.read_csv(ruta)

Clientes.head()


#Revisar número de registros del dataset

len(Clientes)
```

```
Clientes.info()


#Estadísticas de variables numéricas

Clientes.describe()


#Estadísticas de variables categóricas

Clientes.describe(include='O')


#Visualización del conjunto de datos

Clientes.head(10)


#Matriz de correlación

correlation_matrix = Clientes.corr()

fig, ax = plt.subplots(figsize=(12, 10))

sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5, ax=ax, annot_kws={"fontsize": 8})

ax.set_title('Correlation Heatmap')

ax.set_xticklabels(ax.get_xticklabels(), rotation=45, ha='right', fontsize=8)

ax.set_yticklabels(ax.get_yticklabels(), rotation=0, ha='right', fontsize=8)

cbar = ax.collections[0].colorbar

cbar.ax.tick_params(labelsize=8)

plt.tight_layout()

plt.show()
```

```
Clientes.shape

# ANALISIS EXPLORATORIO

# DESCRIPCION GENERAL DE LOS DATOS

# Información general sobre el DataFrame

info_general = Clientes.info()


# Tipos de datos de cada columna

tipos_de_datos = Clientes.dtypes


# Conteo de valores no nulos en cada columna

valores_no_nulos = Clientes.count()


# Conteo de valores únicos en cada columna

valores_unicos = Clientes.nunique()


# Crear un DataFrame resumen

resumen_general = pd.DataFrame({

    'Tipos de Datos': tipos_de_datos,

    'Valores No Nulos': valores_no_nulos,

    'Valores Únicos': valores_unicos

})
```

```
# Imprimir el resumen general

print(resumen_general)


# ANALISIS EXPLORATORIO

# DESCRIPCION GENERAL DE LOS DATOS NUMERICOS


# Descripción estadística de variables numéricas

descripcion_estadistica = Clientes.describe()

print(descripcion_estadistica)


# ANALISIS EXPLORATORIO

# DESCRIPCION GENERAL DE LOS DATOS CATEGORICOS

descripcion_categorica = Clientes.describe(include='O')

print(descripcion_categorica)


#TAMAÑO DEL DATASET

Clientes.shape


#VALORES NULOS

Clientes.isnull().sum()
```

```
#Revisar los datos faltantes en la columna TotalCharges

Clientes[Clientes['TotalCharges']==' ']


# Reemplazar datos faltantes en la columna 'TotalCharges' con la mediana

Clientes['TotalCharges'] = Clientes['TotalCharges'].replace(' ', np.nan)

Clientes['TotalCharges'] = pd.to_numeric(Clientes['TotalCharges'])

Clientes['TotalCharges'].fillna(Clientes['TotalCharges'].median(), inplace=True)


Clientes[Clientes['TotalCharges']==' ']


# Verificar los valores despues de reemplazar TotalCharges con mediana

valores_nulos = Clientes.isnull().sum()

print("\nValores nulos:\n", valores_nulos)


#Revisar estadísticas descriptivas para variables numéricas

estadisticas_num = Clientes.describe()

print("\nEstadísticas descriptivas de variables numéricas:\n", estadisticas_num)


#Revisar valores únicos para variables categóricas

valores_unicos_cat = Clientes.select_dtypes(include='object').nunique()

print("\nValores únicos de variables categóricas:\n", valores_unicos_cat)
```

```
#Revisar valores nulos categóricos

valores_nulos_cat = Clientes.isnull().sum()

print("Valores nulos por columna:\n", valores_nulos_cat)


"""#Preparación de la data"""


Clientes.head()


#Revisar todas las columnas de la data

Clientes.columns


#Realizar una lista de los valores únicos en cada columna de Clientes y se separa visualmente los resultados para cada columna con una línea de guiones.

for column in Clientes.columns:

    print(f"Unique values in '{column}' column:")

    print(Clientes[column].unique())

    print("\n" + "="*40 + "\n")


#Revisar los tipos de datos de cada columna

Clientes.dtypes


#Revisar el dataset

Clientes.head()
```

```
#Seleccionar las variables tenure, TotalCharges y Churn

Clientes.loc[Clientes['tenure'] == 0, ['tenure', 'TotalCharges', 'Churn']]


#Revisar los valores nulos del Dataframe Clientes en una columna adicional

nulls = pd.DataFrame(Clientes.isna().sum())


nulls


#Reemplazar las cantidades de la columna SeniorCitizen

Clientes['SeniorCitizen'].replace({0:'No', 1:'Yes'},inplace=True)


#Recuento de las cantidades de clientes de la columna Churn

Clientes['Churn'].value_counts()


#Ahora se muestra el porcentaje de clientes de la columna Churn al multiplicar por 100

percentage_churn = Clientes['Churn'].value_counts(normalize=True) * 100

print(percentage_churn)


"""#GRÁFICOS DE DISTRIBUCIÓN"""


#COLUMNA GÉNERO
```

```
#Número de personas de tercera edad

Clientes['gender'].value_counts()


#COLUMNA GÉNERO

#Número de Género con masculino y femenino en total

sns.set(style="whitegrid")

plt.figure(figsize=(8, 6))

# Crear el countplot

ax = sns.countplot(x="gender", data=Clientes, palette="Set2")

# Agregar números en las barras

for p in ax.patches:

    height = p.get_height()

    ax.text(p.get_x() + p.get_width()/2., height + 0.1, height, ha="center", fontsize=11)

plt.xlabel('Gender')

plt.ylabel('Count')

plt.title('Distribución por género')

plt.show()


#COLUMNA GÉNERO

#Distribución de las columnas gender y Churn en un diagrama de distribución

sns.set(style="whitegrid")

plt.figure(figsize=(8, 6))
```

```
# Crear el countplot
ax = sns.countplot(x='gender', hue='Churn', data=Clientes, palette="Set2")
# Agregar números en las barras
for p in ax.patches:
    if not pd.isnull(p.get_height()):  # Verificar que el valor no sea nulo
        ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()),
                    ha='center', va='center', fontsize=11, color='black', xytext=(0, 5),
                    textcoords='offset points')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.title('Distribución de Churn por Género')
plt.show()


#COLUMNA TERCERA EDAD
#Número de hombres y mujeres
Clientes['SeniorCitizen'].value_counts()


#COLUMNA TERCERA EDAD
#Número de Género con masculino y femenino en total
sns.set(style="whitegrid")
plt.figure(figsize=(8, 6))
# Crear el countplot
```

```
ax = sns.countplot(x="SeniorCitizen", data=Clientes, palette="Set2")

# Agregar números en las barras

for p in ax.patches:

    height = p.get_height()

    ax.text(p.get_x() + p.get_width() / 2., height + 0.1, f'{int(height)}', ha="center", fontsize=11)

plt.xlabel('SeniorCitizen')

plt.ylabel('Count')

plt.title('Distribución por Tercera Edad')

plt.show()


#Distribución de las columnas Tercera edad y Churn en un diagrama de distribución

sns.set(style="whitegrid")

plt.figure(figsize=(8, 6))


# Crear el countplot

ax = sns.countplot(x='SeniorCitizen', hue='Churn', data=Clientes, palette="Set2")


# Agregar números en las barras

for p in ax.patches:

    ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()),

                ha='center', va='center', fontsize=11, color='black', xytext=(0, 5),

                textcoords='offset points')
```

```
plt.xlabel('SeniorCitizen')

plt.ylabel('Count')

plt.title('Distribución de Churn por SeniorCitizen')

plt.show()


#COLUMNA PAREJA

#Número de clientes que tienen pareja

Clientes['Partner'].value_counts()


#Número de personas con pareja en total

sns.set(style="whitegrid")

plt.figure(figsize=(8, 6))

#Crear el countplot

ax = sns.countplot(x="Partner", data=Clientes, palette="Set2")

# Agregar números en las barras

for p in ax.patches:

    ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()),

                ha='center', va='center', fontsize=11, color='black', xytext=(0, 5),

                textcoords='offset points')


plt.xlabel('Partner')
```

```
plt.ylabel('Count')

plt.title('Distribución total de personas con pareja')

plt.show()


#COLUMNA PARTHER

#Distribución de las columnas Pareja y Churn en un diagrama de distribución

sns.set(style="whitegrid")

plt.figure(figsize=(8, 6))


# Crear el countplot

ax = sns.countplot(x='Partner', hue='Churn', data=Clientes, palette="Set2")


# Agregar números enteros en las barras

for p in ax.patches:

    ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_height()),

                ha='center', va='center', fontsize=11, color='black', xytext=(0, 5),

                textcoords='offset points')


plt.xlabel('Partner')

plt.ylabel('Count')

plt.title('Distribución de Churn que tienen pareja')

plt.show()
```

```
#rango de la cantidad de meses que más clientes eligieron nuestros servicios

Clientes['tenure'].quantile([0,0.25,0.5,0.75,1])

df_tenure = pd.DataFrame(Clientes['tenure'])

print(df_tenure)

boxplot = df_tenure.boxplot(column=['tenure'])

boxplot.plot()

plt.show()


Clientes.groupby(['gender']).mean()['tenure']


import matplotlib.pyplot as plt

promedio_tenure_por_genero = Clientes.groupby(['gender']).mean()['tenure']

ax = promedio_tenure_por_genero.plot(kind='bar')

for p in ax.patches:

    ax.annotate(str(round(p.get_height(), 2)), (p.get_x() + p.get_width() / 2., p.get_height()),

                ha='center', va='center', xytext=(0, 10), textcoords='offset points')


plt.title('Promedio de Tenure por Género')

plt.xlabel('Género')

plt.ylabel('Promedio de Tenure')

plt.show()
```

```
#Adulto mayor

Clientes.groupby(['SeniorCitizen']).mean()['tenure']


import matplotlib.pyplot as plt

promedio_tenure_por_senior = Clientes.groupby(['SeniorCitizen']).mean()['tenure']

ax = promedio_tenure_por_senior.plot(kind='bar')

for p in ax.patches:

    ax.annotate(str(round(p.get_height(), 2)), (p.get_x() + p.get_width() / 2., p.get_height()),

                ha='center', va='center', xytext=(0, 10), textcoords='offset points')


plt.title('Promedio de Tenure por SeniorCitizen')

plt.xlabel('SeniorCitizen')

plt.ylabel('Promedio de Tenure')

plt.show()


#Duracion del contrato

Clientes.value_counts('Contract')


import matplotlib.pyplot as plt

plot = Clientes['Contract'].value_counts().plot(kind='pie', autopct='%1.1f%%',

                                        figsize=(6, 6),
```

```
                                    title='Duración del Contrato')


for idx, value in enumerate(Clientes['Contract'].value_counts()):
    plt.text(idx, 0, f'{value}', ha='center', va='center')


plt.show()


#Promedio de Cargos mensuales agrupados por el tiempo de contrato de cada cliente
Clientes.groupby(['Contract']).mean()['MonthlyCharges']


import matplotlib.pyplot as plt
promedio_cargos_mensuales_por_contrato                                    =
Clientes.groupby(['Contract']).mean()['MonthlyCharges']
colores = ['blue', 'orange', 'green']
ax = promedio_cargos_mensuales_por_contrato.plot(kind='bar', color=colores)
for p in ax.patches:
    ax.annotate(str(round(p.get_height(), 2)), (p.get_x() + p.get_width() / 2., p.get_height()),
                ha='center', va='center', xytext=(0, 10), textcoords='offset points')


plt.title('Promedio de MonthlyCharges por Contrato')
plt.xlabel('Tipo de Contrato')
plt.ylabel('Promedio de MonthlyCharges')
plt.show()
```

```
import matplotlib.pyplot as plt

import seaborn as sns


Info = ['gender', 'SeniorCitizen', 'Partner', 'Dependents']

hue = "Churn"

figsize = (30, 9)

palette2 = ['#FF6347', '#4169E1']


def countplots_custom(dataset, columns_list, rows, cols, figsize, suptitle, hue, palette):

    fig, axs = plt.subplots(rows, cols, sharey=True, figsize=figsize)

    fig.suptitle(suptitle, y=1, size=35)

    axs = axs.flatten()

    for i, data in enumerate(columns_list):

        ax = sns.countplot(data=dataset, ax=axs[i], x=columns_list[i], hue=hue, palette=palette,
edgecolor='black')

        axs[i].set_title(data + f' vs {hue}', size=25)

        for container in ax.containers:

            ax.bar_label(container, size=20)

            ax.set_xlabel('')


Info = ['gender','SeniorCitizen','Partner','Dependents']

hue = "Churn"
```

```
figsize = (30,9)

palette2 = ['#FF6347', '#4169E1']

countplots_custom(dataset=Clientes, columns_list=Info, rows=1, cols=4, figsize=figsize, hue=hue, palette=palette2, suptitle='Información de Clientes vs variable objetivo Churn')

plt.tight_layout()


import matplotlib.pyplot as plt

import seaborn as sns


Serv = ['PhoneService', 'MultipleLines', 'InternetService', 'StreamingTV', 'StreamingMovies']

fig, axes = plt.subplots(2, 3, figsize=(20, 14))


for i, column in enumerate(Serv):

    row = i // 3

    col = i % 3

    ax = axes[row, col]


    sns.countplot(data=Clientes, x=column, hue="Churn", palette=palette2, edgecolor='black', ax=ax)

    ax.legend(loc='upper right' if col == 2 else 'upper left', title='Churn')


    for container in ax.containers:

        ax.bar_label(container)
```

```
    title = f'{column} vs Churn'
    ax.set_title(title)

plt.tight_layout()
plt.show()

Pago =  ['Contract','PaperlessBilling','PaymentMethod']
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(25, 7))
ax = sns.countplot(data=Clientes, x=Pago[0], hue="Churn", palette=palette2, edgecolor='black', ax=axes[0])
ax.legend(loc='upper right', title='Churn')
for i in ax.containers:
    ax.bar_label(i,)
title = Pago[0] + ' vs Churn'
ax.set_title(title)
ax = sns.countplot(data=Clientes, x=Pago[1], hue="Churn", palette=palette2, edgecolor='black', ax=axes[1])
ax.legend(loc='lower left', title='Churn')
for i in ax.containers:
    ax.bar_label(i,)
title = Pago[1] + ' vs Churn'
ax.set_title(title)
```

```
ax = sns.countplot(data=Clientes, x=Pago[2], hue="Churn", palette=palette2, edgecolor='black', ax=axes[2])

ax.legend(loc='lower left', title='Churn')

for i in ax.containers:

    ax.bar_label(i,)

ax.set_xticklabels(['e-check', 'm-check', 'Bank transfer', 'Credit card'])

title = Pago[2] + ' vs Churn'

ax.set_title(title)

plt.tight_layout()

plt.show()


Servicios = ['OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport']

countplots_custom(dataset=Clientes, columns_list=Servicios, rows=1, cols=4, figsize=(25, 8), hue=hue, palette=palette2, suptitle='Soporte de servicios vs variable objetivo (Churn)')

plt.tight_layout()


#Eliminar la columna customerID ya que no es reelevante para este estudio

Clientes = Clientes.drop('customerID', axis=1)


#AGRUPAR POR COLUMNAS CATEGORICAS Y NUMÉRICAS

columns = list(Clientes.columns)


categoric_columns = []
```

```
numeric_columns = []


for i in columns:
    if len(Clientes[i].unique()) >=6:
        numeric_columns.append(i)
    else:
        categoric_columns.append(i)


categoric_columns = categoric_columns[:-1]


Clientes[numeric_columns].describe()


Clientes[categoric_columns].describe()


#VERIFICAR COLUMNAS NUMERICAS
numeric_columns


#DIAGRAMAS DE DENSIDAD PARA VARIABLES CATEGORICAS
def dist_custom(dataset, columns_list, rows, cols, suptitle):
    fig, axs = plt.subplots(rows, cols,figsize=(20,5))
    fig.suptitle(suptitle,y=1, size=25)
    axs = axs.flatten()
```

```
    for i, data in enumerate(columns_list):
        sns.kdeplot(dataset[data], ax=axs[i], fill=True, alpha=0.8, linewidth=0, color='#2E86C1')
        axs[i].set_title(data + ', skewness is '+str(round(dataset[data].skew(axis = 0, skipna = True),2)))

dist_custom(dataset=Clientes, columns_list=numeric_columns, rows=1, cols=5, suptitle='Distribución para cada característica numérica')

plt.tight_layout()


#DIAGRAMAS DE DENSIDAD PARA VARIABLES CATEGORICAS CON BARRAS

def plot_custom_distribution(dataset, columns_list, rows, cols, suptitle):
    fig, axs = plt.subplots(rows, cols, figsize=(20,5))
    fig.suptitle(suptitle, y=1, size=25)
    axs = axs.flatten()
    for i, data in enumerate(columns_list):
        sns.histplot(dataset[data], ax=axs[i], kde=True, fill=True, alpha=0.8, color='#2E86C1')
        axs[i].set_title(f"{data}, skewness: {round(dataset[data].skew(skipna=True), 2)}")
        axs[i].set_xlabel('')
    plt.tight_layout()

plot_custom_distribution(dataset=Clientes, columns_list=numeric_columns, rows=1, cols=5, suptitle='Distribución para cada característica numérica con diagramas de barras')

plt.show()


#Paleta de colores

palette = ['#008080','#FF6347', '#E50000', '#D2691E']
```

```
palette2 = ['#FF6347', '#008080', '#E50000', '#D2691E']


#REVISAR CON BOXPLOTS

def boxplots_custom(dataset, columns_list, rows, cols, suptitle, palette='Blues'):
    fig, axs = plt.subplots(rows, cols, sharey=True, figsize=(20, 4))
    fig.suptitle(suptitle, y=1, size=25)
    axs = axs.flatten()
    for i, data in enumerate(columns_list):
        sns.boxplot(data=dataset[data], orient='h', ax=axs[i], palette=palette)
        axs[i].set_title(f"{data}, skewness is: {round(dataset[data].skew(skipna=True), 2)}")
    plt.tight_layout()
boxplots_custom(dataset=Clientes, columns_list=numeric_columns, rows=1, cols=5, suptitle='Boxplots for numerical features')

plt.show()


#VISUALIZACION DE CLIENTES QUE CANCELARON Y NO CANCELARON EL SERVICIO


def plot_churn_data(data):
    l1 = list(data['Churn'].value_counts())
    pie_values = [l1[0] / sum(l1) * 100, l1[1] / sum(l1) * 100]


    fig, axes = plt.subplots(nrows=1, ncols=2, figsize=(15, 5))
```

```
plt.subplot(1, 2, 1)

palette_pie = sns.color_palette("Set2")

plt.pie(pie_values, labels=['Clientes que no abandonaron', 'Clientes que abandonaron'],
        autopct='%1.2f%%',
        explode=(0.1, 0),
        colors=palette_pie,
        wedgeprops={'edgecolor': 'black', 'linewidth': 1, 'antialiased': True})
plt.title('Clientes que abandonaron y no abandonaron el servicio')


plt.box(on=True)


plt.subplot(1, 2, 2)

palette_count = sns.color_palette("Set2")

ax = sns.countplot(data=data,
                   x='Churn',
                   palette=palette_count,
                   edgecolor='black')
for i in ax.containers:
    ax.bar_label(i)
ax.set_xticklabels(['Clientes que no abandonaron', 'Clientes que abandonaron'])


plt.title('Clientes que abandonaron y no abandonaron el servicio')
```

```
    plt.box(on=True)


    plt.tight_layout()

    plt.show()


# Llama a la función plot_churn_data con tus datos (Clientes)

plot_churn_data(Clientes)


"""ANALISIS DE EXPLORACIÓN DE DATOS"""


#REEMPLAZAR VALORES REDUNDANTES QUE TIENEN EL CAMPO NO

Clientes.replace({'No phone service':'No','No internet service':'No'},inplace=True)


#REVISAR EL NUEVO DATASET

Clientes.head()


#Se crea un nuevo dataset yes_no_columns que contiene el nombre de las columnas del DataFrame Base donde todos los valores son 'Yes' y 'No'

def has_yes_no_values(column):

    unique_values = column.unique()

    return set(unique_values) == {'Yes', 'No'}


#Identifico columnas con valores "Yes" y "No"
```

```
yes_no_columns = [col for col in Clientes.columns if has_yes_no_values(Clientes[col])]


#Despliega columnas con valores "Yes" y "No"

print("Columns with 'Yes' and 'No' values:", yes_no_columns)


#Mostrar las columnas

Clientes.columns


#Cambio para columnas tienen exclusivamente valores 'Yes' o 'No',

Yes_No_Columns = ['SeniorCitizen', 'Partner', 'Dependents', 'PhoneService', 'PaperlessBilling', 'Churn']


#Se convierte variables categóricas en valores numéricas, asignando un número entero a cada categoría única.

from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()

for column in Yes_No_Columns:

    Clientes[column] = label_encoder.fit_transform(Clientes[column])


Clientes.head()


Clientes.columns
```

```
#Se convierte variables categóricas en un formato numérico

columns_to_encode = ['gender','MultipleLines', 'InternetService', 'OnlineSecurity','OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract','PaymentMethod' ]


Clientes.head(5)


#Se convierte en Dummieslas columnas

for column in columns_to_encode:

  column_dummies = pd.get_dummies(Clientes[column], prefix=f'{column}_', dummy_na=False)

  Clientes = pd.concat([Clientes, column_dummies], axis=1)


Clientes.head(5)


Clientes.columns


#Eliminar las columnas especificadas en la lista columns_to_encode porque ya fueron tranformadas con dummies

Clientes = Clientes.drop(columns=columns_to_encode)


Clientes.head()


Clientes.columns
```

```
import pandas as pd

from sklearn.preprocessing import StandardScaler

columns_to_scale = [col for col in Clientes.columns if col != 'Churn']

Clientes_selected = Clientes[columns_to_scale]

churn_column = Clientes['Churn']

scaler = StandardScaler()

Clientes_st_sc = scaler.fit_transform(Clientes_selected)

Clientes_st_sc_df = pd.DataFrame(Clientes_st_sc, columns=columns_to_scale)

Clientes_st_sc_df['Churn'] = churn_column

print(Clientes_st_sc_df)


# Display the resulting DataFrame

Clientes_st_sc_df.head()


"""#DECLARACIÓN DE CONJUNTO DE PARAMETROS DE TRAIN Y TEST"""


from sklearn.model_selection import train_test_split

#ELIMINACION LA COLUMNA CHURN Y REEMPLAZO POR 1 Y 0

X = Clientes_st_sc_df.drop('Churn', axis=1)

y = Clientes_st_sc_df['Churn']
```

```
#CONJUNTO DE ENTRENAMIENTO Y PRUEBA

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3,

                                   random_state=1, stratify=y)


churn_column


Clientes['Churn'].value_counts()


#IMPRIMIR FORMAS DE LOS CONJUNTOS DE DATOS DE ENTRENAMIENTO Y PRUEBAS

print('X shape\t\t:', X.shape)

print('y shape\t\t:', y.shape)

print()

print('X_train shape\t:', X_train.shape)

print('y_train shape\t:', y_train.shape)

print()

print('X_test shape\t:', X_test.shape)

print('y_test shape\t:', y_test.shape)


"""#1) Algoritmo Regresión Logística"""


from sklearn import linear_model

Classifier = linear_model.LogisticRegression()
```

```
Classifier.fit(X_train, y_train)

# make predictions for test data

y_pred = Classifier.predict(X_test)


print(list(y_test))

print(list(y_pred))


value_counts_series = pd.Series(y_test).value_counts()


value_counts_series


value_counts_series_1 = pd.Series(y_pred).value_counts()


value_counts_series_1


from sklearn.metrics import confusion_matrix

from sklearn.metrics import accuracy_score

# Evaluando predicciones

cm = confusion_matrix(y_test, y_pred)

print(cm)

accuracy=accuracy_score(y_test,y_pred)

reg_logistica=("%.2f%%" % (accuracy * 100.0))
```

```
print(reg_logistica)


from sklearn.linear_model import LogisticRegression


modeloLog = LogisticRegression()

modeloLog.fit(X_train, y_train)

scoreLog = modeloLog.score(X_test,y_test)

print(scoreLog)


print(classification_report(y_test, prediccionesLog))


from sklearn.metrics import confusion_matrix, accuracy_score

import seaborn as sns

import matplotlib.pyplot as plt


# Evaluando predicciones

cm = confusion_matrix(y_test, y_pred)

print("Matriz de Confusión:")

print(cm)


accuracy = accuracy_score(y_test, y_pred)

reg_logistica = ("%.2f%%" % (accuracy * 100.0))
```

```
print("Precisión:", reg_logistica)


# Obtener nombres de clases únicas

unique_classes = sorted(set(y_test))


# Visualización de la matriz de confusión con un mapa de calor

sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=unique_classes, yticklabels=unique_classes)

plt.xlabel('Predicciones')

plt.ylabel('Etiquetas reales')

plt.title('Matriz de Confusión')

plt.show()


prediccionesLog = modeloLog.predict(X_test)

prediccionesLog


confusion_matrix = metrics.confusion_matrix(y_test, y_pred)

cm_display=metrics.ConfusionMatrixDisplay(confusion_matrix
=confusion_matrix,display_labels = [False, True])


cm_display.plot()

plt.show()
```

```
from sklearn.metrics import confusion_matrix


# Calculate confusion matrix

conf_matrix = confusion_matrix(y_test, y_pred)


# Extract values from confusion matrix

tn, fp, fn, tp = conf_matrix.ravel()


# Calculate accuracy

accuracy = (tp + tn) / (tp + tn + fp + fn)


# Calculate precision

precision = tp / (tp + fp)


# Calculate recall

recall = tp / (tp + fn)


specificity = tn / (tn + fp)


# Calculate F1 score

f1 = 2 * (precision * recall) / (precision + recall)
```

```
# Print the metrics

print("Accuracy:", accuracy)

print("Precision:", precision)

print("Recall:", recall)

print("Specificity:", specificity)

print("F1 Score:", f1)


from sklearn.metrics import roc_curve, auc

import matplotlib.pyplot as plt


fpr, tpr, lim = roc_curve(y_test, y_pred)

roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='red', lw=2, label=f'Curva ROC (AUC = {roc_auc:.2f})')

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel("Tasa de Falsos Positivos")

plt.ylabel("Tasa de Verdaderos Positivos")

plt.title("Curva ROC")

plt.legend(loc="lower right")

plt.show()
```

```
"""#2) ARBOL DE DECISIÓN"""


from sklearn import tree

arbol = tree.DecisionTreeClassifier(

    criterion = 'entropy',

    max_depth= 5,

    min_samples_split=2,

    min_samples_leaf = 1

)

arbol = arbol.fit(X_train, y_train)


tree.plot_tree(arbol, filled = True)

plt.show()


# Predicciones

predicciones = arbol.predict(X_train)

print(predicciones)

# Comparativa

exactitud = accuracy_score(y_train,predicciones)

print(exactitud)

# Matriz de confusión
```

```
matriz_conf = confusion_matrix(y_train, predicciones)

print(matriz_conf)


sns.set()

#Generación de clases

classes = ['Si', 'No']

sns.heatmap(matriz_conf, annot=True, fmt = 'd', cmap = 'Blues', xticklabels=classes,

          yticklabels= classes)

plt.title("Matriz de confusión")

plt.xlabel("Etiqueta predicha")

plt.ylabel("Etiqueta Real")

plt.show()


exactitud = accuracy_score(y_train,predicciones)

print('Exactitud: ',exactitud*100)

precision = precision_score(y_train,predicciones)

print('Precisión: ', precision*100)

sensibilidad = recall_score(y_train,predicciones)

print('Recall: ', sensibilidad*100)

puntaje = f1_score(y_train,predicciones)

print('Puntaje: ', puntaje*100)

matriz_Tree = confusion_matrix(y_train, predicciones)
```

```
matriz_Tree


print(classification_report(y_train,predicciones))


#CURVA ROC
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
fpr, tpr, lim = roc_curve(y_train, predicciones)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='red', lw=2, label=f'Curva ROC (AUC = {roc_auc:.2f})')  # Ajuste en la etiqueta
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel("Tasa de Falsos Positivos")
plt.ylabel("Tasa de Verdaderos Positivos")
plt.title("Curva ROC en Datos de Entrenamiento")
plt.legend(loc="lower right")
plt.show()


"""#3) REDES NEURONALES"""
```

```
from sklearn.neural_network import MLPClassifier

red = MLPClassifier(random_state=1, max_iter=300).fit(X_train, y_train)


predicciones_nn = red.predict(X_test)

# Comparativa

exactitud = accuracy_score(y_test,predicciones_nn)

redes_exactitud=exactitud

print(redes_exactitud)

# Matriz de confusión

matriz_conf = confusion_matrix(y_test, predicciones_nn)

print(matriz_conf)


from sklearn.metrics import classification_report

y_pred = red.predict(X_test)

# Imprimir el informe de clasificación

print(classification_report(y_test, y_pred))


classes = ['Si', 'No']

sns.heatmap(matriz_conf, annot=True, fmt = 'd', cmap = 'Blues', xticklabels=classes,

          yticklabels= classes)

plt.title("Matriz de confusión")

plt.xlabel("Etiqueta predicha")
```

```
plt.ylabel("Etiqueta Real")

plt.show()


#CURVA ROC

fpr, tpr, lim = roc_curve(y_test, predicciones_nn)

roc_auc = auc(fpr, tpr)

plt.figure(figsize=(8, 6))

plt.plot(fpr, tpr, color='red', lw=2, label=f'Curva ROC (AUC = {roc_auc:.2f})')  # Ajuste en la etiqueta

plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel("Tasa de Falsos Positivos")

plt.ylabel("Tasa de Verdaderos Positivos")

plt.title("Curva ROC en Datos de Prueba")

plt.legend(loc="lower right")

plt.show()


"""# 4) RANDOM FOREST CLASSIFIER"""


from sklearn.ensemble import RandomForestClassifier


# Crear el modelo con 100 arboles
```

```
model = RandomForestClassifier(n_estimators=100,
                               bootstrap = True, verbose=2,
                               max_features = 'sqrt')
# a entrenar!
model.fit(X_train, y_train)


#RandomForestClassifier
forest = RandomForestClassifier()
forest.fit(X_train, y_train)


#REALIZAR PREDICCIONES
y_pred_test = forest.predict(X_test)


#VER EL ACCURACY
accuracy_score(y_test, y_pred_test)


#IMPRIMIENDO LA PREDICCION
random=(metrics.accuracy_score(y_test, y_pred_test))
print (random)


svc = SVC(random_state=42)
svc.fit(X_train, y_train)
```

```
svc_disp = RocCurveDisplay.from_estimator(svc, X_test, y_test)

plt.show()


#CURVA ROC

rfc = RandomForestClassifier(n_estimators=10, random_state=42)

rfc.fit(X_train, y_train)

ax = plt.gca()

rfc_disp = RocCurveDisplay.from_estimator(rfc, X_test, y_test, ax=ax, alpha=0.8)

svc_disp.plot(ax=ax, alpha=0.8)

plt.show()


#MATRIZ DE CONFUSION DE LAS VARIABLES PREDICTORAS Y DE ENTRENAMIENTO

conf_matrix =confusion_matrix(y_test, y_pred_test)

confusion_matrix(y_test, y_pred_test)


# View the classification report for test data and predictions

print(classification_report(y_test, y_pred_test))


data = {

    'precision': [0.89, 0.75],

    'recall': [0.92, 0.68],
```

```
    'f1-score': [0.90, 0.71],
    'support': [1552, 561]
}

# Crear un DataFrame de pandas
index = ['0', '1']
results_df = pd.DataFrame(data, index=index)

# Agregar la fila 'accuracy'
results_df.loc['accuracy'] = [0.86, 0.86, 0.86, 2113]

# Mostrar la tabla de resultados
print(results_df)

classes = ['Si', 'No']
sns.heatmap(conf_matrix, annot=True, fmt = 'd', cmap = 'Blues', xticklabels=classes,
            yticklabels= classes)
plt.title("Matriz de confusión")
plt.xlabel("Etiqueta predicha")
plt.ylabel("Etiqueta Real")
plt.show()
```

```
"""#5) LAZY CLASSIFIER"""


from lightgbm import LGBMClassifier

# Create the LGBMClassifier

lgbm_classifier = LGBMClassifier()

# Fit the classifier to the training data

lgbm_classifier.fit(X_train, y_train)

# Make predictions on the test data

y_predL = lgbm_classifier.predict(X_test)

# Evaluate the accuracy of the classifier

accuracy = accuracy_score(y_test, y_predL)

print("Accuracy:", accuracy)


# Confusion matrix

cm_algorithm1 = confusion_matrix(y_test, y_predL)

plt.figure(figsize=(8, 6))

sns.heatmap(cm_algorithm1, annot=True, fmt="".0f", cmap='Blues')

plt.title('Confusion Matrix - Lazy Classifier')

plt.xlabel('Predicted Labels')

plt.ylabel('True Labels')

plt.show()
```

```
from lightgbm import LGBMClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, f1_score

lgbm_classifier = LGBMClassifier()

lgbm_classifier.fit(X_train, y_train)

y_predL = lgbm_classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_predL)

print("Accuracy:", accuracy)

conf_matrix = confusion_matrix(y_test, y_predL)

print("Confusion Matrix:\n", conf_matrix)

class_report = classification_report(y_test, y_predL)

print("Classification Report:\n", class_report)

f1 = f1_score(y_test, y_predL)

print("F1 Score:", f1)


from sklearn.metrics import roc_curve, roc_auc_score

# Calcular las probabilidades de las clases

y_pred_prob = model.predict_proba(X_test)

# Calcular la tasa de verdaderos positivos, tasa de falsos positivos y umbrales usando roc_curve

fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob[:,1])

# Calcular el área bajo la curva ROC

roc_auc = roc_auc_score(y_test, y_pred_prob[:,1])
```

```
# Graficar la curva ROC
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='red', label='ROC curve (AUC = {:.2f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend()
plt.show()


#IMPRIMIENDO RESULTADOS
print('Regresión Logística', reg_logistica)
print('Arbol de decisión: ',exactitud*100)
print('Redes Neuronales:', redes_exactitud)
print ('Random Forest Classifier: ',random)
print("Lazy Classifier:", accuracy)


#IMPRIMIENDO RESULTADOS EN UNA TABLA
from tabulate import tabulate
results = [
    ['Regresión Logística',reg_logistica],
     ['Arbol de decisión', exactitud*100],
```

```
    ['Redes Neuronales', redes_exactitud],
    ['Random Forest Classifier', random],
    ['Lazy Classifier', accuracy]
]
# Imprimir la tabla con estilo 'grid'
print(tabulate(results, headers=['MODELO PREDICTIVO', 'PRECISIÓN APROXIMADA'], tablefmt='grid'))


from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neural_network import MLPClassifier
from lightgbm import LGBMClassifier
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns


log_reg = LogisticRegression()
decision_tree = DecisionTreeClassifier()
neural_network = MLPClassifier()
random_forest = RandomForestClassifier()
lgbm_classifier = LGBMClassifier()
```

```
models = [log_reg, decision_tree, random_forest, lgbm_classifier]

model_names = ['Logistic Regression', 'Decision Tree', 'neural_network','Random Forest','Lazy Classifier']


n_models = len(models)

n_cols = 2

n_rows = n_models // n_cols + (n_models % n_cols > 0)


plt.figure(figsize=(10 * n_cols, 8 * n_rows))


for i, (model, name) in enumerate(zip(models, model_names)):

    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)


    cm = confusion_matrix(y_test, y_pred)


    plt.subplot(n_rows, n_cols, i + 1)

    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

    plt.title(f'Matriz de Confusión para {name}')

    plt.xlabel('Predicción')

    plt.ylabel('Verdadero')
```

```
plt.tight_layout()

plt.show()


from sklearn.metrics import f1_score, recall_score, precision_score, roc_curve, auc, roc_auc_score

from sklearn.metrics import f1_score, recall_score, precision_score, roc_curve, auc

import pandas as pd

f1_scores = {}

recall_scores = {}

precision_scores = {}

roc_auc_scores = {}


plt.figure(figsize=(10, 6))


# Entrenar, evaluar y calcular métricas para cada modelo

for i, (model, name) in enumerate(zip(models, model_names)):

    # Haciendo predicciones con el modelo

    y_pred = model.predict(X_test)


    # Calculando métricas

    f1_scores[name] = f1_score(y_test, y_pred)

    recall_scores[name] = recall_score(y_test, y_pred)

    precision_scores[name] = precision_score(y_test, y_pred)
```

```
    # Calculando la curva ROC y el área bajo la curva si el modelo soporta probabilidades
    if hasattr(model, "predict_proba"):
        y_pred_proba = model.predict_proba(X_test)[:, 1]
        fpr, tpr, _ = roc_curve(y_test, y_pred_proba)
        roc_auc = auc(fpr, tpr)
        roc_auc_scores[name] = roc_auc

        # Agregar la curva ROC al gráfico
        plt.plot(fpr, tpr, label=f'{name} (AUC = {roc_auc:.2f})')

# Finalizando la gráfica de la curva ROC
plt.plot([0, 1], [0, 1], 'k--', label='Random Chance')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curves')
plt.legend(loc='lower right')
plt.show()

f1_scores_table = pd.DataFrame(f1_scores.items(), columns=['Model', 'F1 Score'])
recall_scores_table = pd.DataFrame(recall_scores.items(), columns=['Model', 'Recall Score'])
precision_scores_table = pd.DataFrame(precision_scores.items(), columns=['Model',
```

```
'Precision Score'])


# Función para imprimir tablas con bordes utilizando tabulate
def print_table_with_border(df, title):
    print(f"{'=' * 10} {title} {'=' * 10}")
    print(tabulate(df, headers='keys', tablefmt='pretty'))
    print('\n')


# Mostrar los DataFrames como tablas con bordes
print_table_with_border(f1_scores_table, "F1 Scores")
print_table_with_border(recall_scores_table, "Recall Scores")
print_table_with_border(precision_scores_table, "Precision Scores")
#IMPRIMIENDO RESULTADOS EN UNA TABLA
from tabulate import tabulate
results = [
    ['Regresión Logística',reg_logistica],
    ['Arbol de decisión', exactitud*100],
    ['Redes Neuronales', redes_exactitud],
    ['Random Forest Classifier', random],
    ['Lazy Classifier', accuracy]
]
print(tabulate(results, headers=['MODELO PREDICTIVO', 'PRECISIÓN'], tablefmt='grid'))
```